Neural Techniques for Modeling Visually Grounded Speech

by

Kenneth Leidal

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2018

Author . . . .
Department of Electrical Engineering and Computer Science
June 8, 2018

Certified by. . . . .
James Glass
Senior Research Scientist
Thesis Supervisor

Certified by. . . . .
David Harwath
Research Scientist
Thesis Supervisor

Accepted by . . . .
Katrina LaCurts
Undergraduate Officer

Neural Techniques for Modeling Visually Grounded Speech

by

Kenneth Leidal

Submitted to the Department of Electrical Engineering and Computer Science
on June 8, 2018, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

In this thesis, I explore state of the art techniques for using neural networks to learn semantically-rich representations for visual and audio data. In particular, I analyze and extend the model introduced by Harwath et al. (2016), a neural architecture which learns a non-linear similarity metric between images and audio captions using sampled margin rank loss. In Chapter 1, I provide a background on multimodal learning and motivate the need for further research in the area. In addition, I give an overview of Harwath et al. (2016)'s model, variants of which will be used throughout the rest of the thesis. In Chapter 2, I present a quantitative and qualitative analysis of the modality retrieval behavior of the state of the art architecture used by Harwath et al. (2016), identifying a bias towards certain examples and proposing a solution to counteract that bias. In Chapter 3, I introduce the property of modality invariance and explain a regularization technique I created to promote this property in learned semantic embedding spaces. In Chapter 4, I apply the architecture to a new dataset containing videos, which offers unique opportunities to include temporal visual data and ambient audio unavailable in images. In addition, the video domain presents new challenges, as the data density increases with the additional time dimension. I conclude with a discussion about multimodal learning, language acquisition, and unsupervised learning in general.

Thesis Supervisor: James Glass
Title: Senior Research Scientist

Thesis Supervisor: David Harwath
Title: Research Scientist

Acknowledgments

There are so many people I have to thank for their help and support during the course of writing this thesis. First and foremost, my advisors and mentors, David Harwath and James Glass. I noticed that I tend to brainstorm a lot of ideas, some of them good, many of them bad. Dave and Jim were there to help me filter my stream of ideas, knowing which ones had already been tried, which ones were likely to fail, and which ones had potential. They helped me keep focused on areas which showed promise and which I pursued until I found results. They recognized the merit behind my idea of modality invariance (Chapter 3) and co-authored a conference paper on it with me (Leidal et al. 2017). Dave's Ph.D. thesis was the foundation which I built upon for all of my models, and even as he was busy finishing up his thesis and publishing papers of his own, he had time to chat and discuss ideas with me, celebrating my highs and giving me suggestions on where to look next at my lows. Dave has a passion and excitement that rubs off on everyone who has the pleasure of working with him. The same goes for Jim. I couldn't have asked for a better advisor; Jim is patient and kind, and he knows how to guide his students to do their best work without being overwhelmed by it.

I would also like to thank all of my lab mates, but particularly Wei-Ning Hsu. Wei-Ning was always a helpful person to discuss ideas with. He is one of the smartest people I know and can learn new things incredibly fast. He also helped introduce me to people at the first conference I attended where I didn't know anyone, for which I am extremely grateful.

To my roommates, Matt and Karan, thank you for putting up with my messy habits at some of the more stressful points of the year and for pulling some of my weight when I had to prioritize thesis.

My girlfriend, Lauren, was a constant source of support. She had to put up with my ups and downs behind the scenes. She was an advocate for my mental and physical health when I was too distracted to care for myself properly. I'm looking forward to a much needed trip to Maine with her to unwind while hiking in the mountains.

My friend, Carter, pushed me beyond my limits, encouraging me to take difficult classes outside my comfort zone that in hindsight were very worthwhile. He was a great group project partner and was always there to bounce ideas off of.

I wouldn't be where I am without my family. My parents, Sara and Knute, and my brother, Erik, have supported and loved me unconditionally through my years of extreme focus and dedication to my schoolwork. In addition, if my parents, who are both software engineers, hadn't introduced me to the world of programming in elementary school, I doubt I would be the MIT computer science graduate I am today. My grandparents, Jane, Ken, Knute, and Betty have served as a source of inspiration and support for me throughout my years of school.

Finally, without Marcia Davidson, the lab would come to a screeching halt, and Anne Hunter is the glue that holds the department together and makes the course 6 experience better for everyone.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Artificial Intelligence?

I had my first exposure to machine learning three years ago while doing a Tensorflow tutorial writing a convolutional neural network (CNN) to classify 32x32 color images in the CIFAR-10 dataset (Abadi et al. 2015; Krizhevsky et al.). CIFAR-10 is a 10 way classification task. In other words, I trained a model to answer the question “Is this image an airplane, automobile, bird, cat, deer, dog, frog, horse, ship, or truck?” nudging its distribution over the 10 labels to be closer and closer to the observed label through the process of stochastic gradient descent. This is the same form many supervised classification tasks take in machine learning. Supervised, meaning, there are pairs of inputs and outputs—usually assigned by human annotators—used to train the model.

After I trained my CIFAR-10 classifier, I remember running the network on an image of my cat; the output: "90% cat, 10% horse". At the time, that was an exciting result, but now it leaves me dissatisfied. I am not dissatisfied because the network was not 99% certain it was a cat in the image, but rather that it did not understand cats and horses enough to know that they are nothing alike. If that network had been confused about a cat and dog, that I could understand, but a cat and horse, not so much. The network lacked a conceptual understanding of cats and horses. How can we be proud taking a world infinitely rich in visual concepts and reducing that infinity to 10,000, 1,000, or even 10 classes of objects and claim that a model that can correctly identify the objects a significant percentage of the time is intelligent?

Moreover, the process of training a supervised classifier provides much more information to the learner than the way a newborn child learns to recognize objects in the world. When a child points to a cat and says "horse", the mother might correct the child: "*no*, that's a *cat*". But learning from this scenario not only requires a knowledge of the visual stimulus being perceived, but also an ability to understand the mother's feedback: the segmentation of the continuous audio signal into words, the emphasis placed on the words "no" and "cat", and the realization that "cat" is a fundamentally different and potentially new concept than the one that had been perceived: "horse". In other words, supervised learning early in life appears implausible because the labels, themselves, are only available to the learner as noisy sensory inputs.

In contrast to supervised learning, unsupervised learning seeks to find structure in otherwise unlabeled, unstructured data. For example, one might learn new words based on similar sounding unknown words that have previously been encountered (Zhang and Glass 2009; Lee and Glass 2012). Recently, new research has begun in the field of multimodal unsupervised learning (Harwath et al. 2016; 2018b; Harwath and Glass 2017; Harwath et al. 2018a; Leidal et al. 2017). Here, multimodal is taken to mean: concerning two modalities, such as an image and audio of someone verbally describing the image. Like the mother telling the child that the object being perceived is a "cat", the co-occurrence of the sounds producing the word "cat" and the visual image of the "cat" might provide the grounding necessary for a learner to learn a new, previously unrecognized concept.

This thesis explores this unsupervised setting of machine learning. More specifically, I extend a model proposed by Harwath et al. (2016) which is capable of learning semantic concepts through paired images and spoken captions of the images. My efforts focus primarily on exploring new techniques to regularize and transform the semantic space learned to reduce bias, remove noise, and obtain other properties considered desirable for concept learning. In addition, I apply the model to new areas, including a dataset containing short three second video clips (Monfort et al. 2018), showing that the content of the dataset used for training can influence the kinds of words and concepts learned.

1.2 Background

In this section I give an overview of the fundamental concepts required to understand my contributions.

1.2.1 Unsupervised Learning

Unsupervised learning is a promising area of machine learning research whereby models are trained without using manually annotated labels from humans. In contrast to traditional automatic speech recognition (ASR) systems which require a large amount of manually annotated data—from pronunciation dictionaries to transcripts of recorded audio—recent unsupervised models aim to learn meaningful structure from audio alone (Lee and Glass 2012) or audio paired with other sensory inputs (Harwath et al. 2016; Clark and Brennan 1991): data which can be attained by collecting parallel streams of data from sensors.

Example applications of such unsupervised models are (a) pattern recognition to identify the fundamental phonetic building blocks of speech (Lee and Glass 2012), (b) semantic concept learning using an additional modality (vision) to provide grounding traditionally provided by labels (Harwath et al. 2016), and (c) denoising and artificial noise augmentation (Hsu et al. 2017a). Each of these examples involves learning a latent embedding space rich in phonetic, semantic, and acoustic information, respectively.

1.2.2 Multimodal Learning

Multimodal learning involves modeling two or more different channels of sensory input, or modalities. In this thesis, I solely focus on sight and sound, and usually

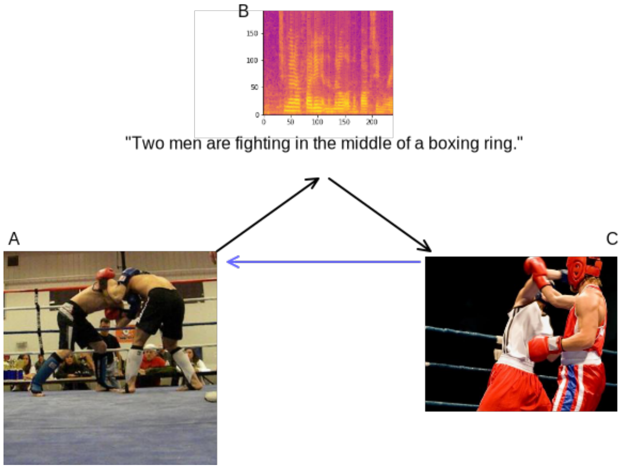


Figure 1-1: Semantically similar, but visually distinct images transitively linked by grounding to the audio caption modality

sound containing speech, but the idea of "multimodal" learning is more general than that. The techniques explored in this thesis may very well apply to other sets of modalities with only minor modifications.

What kinds of tasks could multimodal learning be used for? For one, adding in input from another modality can be used in supervised settings. For instance, in Monfort et al. (2018), the authors find that adding ambient audio in a video action-classification task improves classification performance. The task of modality translation is often referred to by the specific direction of the translation: speech recognition/speech to text, text to speech, image captioning (image to text), conditional image generation (text to image), etc. Modality translation is useful from a practical sense for those who are disabled, lacking the ability to sense a particular modality. Modality retrieval involves finding the most similar instance of one modality given a query in a different modality, by some notion of similarity. For probabilistic formulations of modality translation, a model trained to performed modality translation can be used to perform modality retrieval by selecting the answer with the greatest probability of being generated given the query.

One benefit of multimodal modeling is the potential for the additional modality to provide contextual grounding for extracting semantic information via transitive relationships in the semantic similarity space. For example, see Figure 1-1. If image A is similar to audio B, and audio B is similar to image C, then image A is likely similar to image C by some higher level semantic notion of relevance, even if images A and C are not similar in appearance. This could be especially useful for learning semantic concepts with wide variability in their expression at the sensory level. For instance, there can be wide variability in speech audio due to the speaker's gender or origin, whether the speaker is a native speaker, ambient noise in the room, and even the type of microphone and distance from the microphone to the sound source. Without a higher level notion of semantic similarity relating two instances of a spoken word from different speakers, it could be very difficult for a traditional unsupervised approach, such as segmental dynamic-time-warping or fully Bayesian Gaussian mixture models (Zhang and Glass 2009; Lee and Glass 2012), to recognize the two instances as the same word from the audio signal alone.

1.2.3 Neural Networks

Neural networks are a class of functions which compute their outputs via a system of smaller, interconnected functional units called neurons. They are often trained through stochastic gradient ascent/descent, perturbing the parameters of the network in the direction of the gradient of the objective with respect to the parameters. More precisely, if the goal is to $\max_\theta \mathbb{E}_{x\sim\mathcal{D}} \text{Objective}(x, f(x, \theta_t))$ where x are the inputs (or input/output pairs if supervised) drawn from dataset $\mathcal{D}$, a neural network f with parameters θ can be trained iteratively as follows:

$$\theta_{t+1} = \theta_t + \eta \mathbb{E}_{x\sim\mathcal{D}} \left[\frac{\partial \text{Objective}(x, f(x, \theta_t))}{\partial \theta_t} \right]$$

where η is the step size, a tunable hyperparameter.

The basic building block of the neural network is matrix multiplication. Linear layers involve the matrix multiplication of a matrix of parameters (W, the weight matrix) with a vector of inputs or intermediate "hidden" states and often include a bias vector, b, of parameters:

$$h^{(l+1)} = h^{(l)}W + b$$

Convolutional neural networks (CNNs) operate similarly to a linear layer, but can use input cells surrounding the center cell spatially or temporally as additional input to the matrix multiplication. This spatial/temporal context can be useful for images or audio where local patterns are important for the overall task. The context window is then convolved across the input spatially/temporally, generating an output feature map. The context window is moved a certain number of spaces each iteration, called the "stride". It is important to note that the same parameters, W (often referred to as the "kernel" for CNNs) and b, are shared for each context window. If the kernel size and stride of the convolution is one, the convolution is equivalent to a pixel-wise linear layer between input and output channels.

1.2.4 Speech Recognition and Understanding

Despite advances in recent years leading to the creation of consumer speech recognition products like Apple's Siri, Google's Google/Home, and Amazon's Alexa, speech recognition and understanding is far from a solved problem. Human annotated speech data for training supervised systems is expensive to collect, especially for uncommon languages. In addition, variance between speakers, recording environments, microphones, and the distance and angle between the microphone and sound source can all change signal enough to cause significant error rates in a speech recognition system if not addressed properly. To perform well despite these noise conditions, current top-performing speech recognition systems require a large system of interworking components. Traditionally, GMMs or DNNs model parts of phones in short time scale ($\sim$ 25 milliseconds) intervals called frames, HMMs recognize short-term temporal relationships between frames that represent phones, and a system of composed finite state transducers abstract from phone to phoneme to words. Recent work in supervised speech recognition has looked to using end-to-end neural networks to map speech directly to phones or characters (Graves et al. 2013), but in general end-to-end systems still fail to outperform state-of-the-art GMM/ANN-HMM-FST systems. The expense and difficulty of collecting labeled data for uncommon languages[1] and noise conditions warrants further study into areas of domain adaptation and unsupervised speech understanding.

Speech understanding involves training a model to recognize the semantic content a speaker is attempting to convey. In general, this goal can be difficult to quantify and evaluate, but in specific scenarios like action-oriented reinforcement learning or modality retrieval, there are specific metrics to evaluate the model's understanding.

1.3 Previous Work

The unsupervised learning of semantic relations through the co-occurrence and lack of co-occurrence of sensory inputs is an increasingly attractive pursuit for researchers (Harwath et al. 2016; Wang et al. 2016; Saito et al. 2016; Aytar et al. 2016). This interest is primarily due to the expense of attaining labels for data. The ability to learn semantic relevance with input pairings alone unlocks the potential of training models using inexpensively-collected data with the only supervisory signal being the co-occurrence of sensory inputs (Wang et al. 2016).

In addition, the learned semantic space has direct practical applications. One particular application of a semantic space is cross-modality transfer learning: using paired inputs from two modalities and labels for one modality to learn how to predict labels for the unlabeled modality. Aytar et al. (2016) use a teacher-student model on videos to transfer knowledge from pretrained ImageNet and Places convolutional neural networks (CNNs) identifying object and scene information in images to train a CNN run on the raw audio waveform from the video to recognize the same information. In Aytar et al. (2016)'s model, the shared semantic space consists of the two categorical distributions over objects and scenes as opposed to being a high dimensional Hilbert space, as is the case in the models I explore.

[1] and the impossibility for oral languages

Wang et al. (2016) gave a comprehensive overview of existing approaches to another practical application of shared semantic spaces: cross-modality information retrieval. The task is formulated as follows: given an input of one modality, find related instances of another modality. One traditional approach to solving this problem is to perform canonical correlation analysis on vector representations of paired speech and audio to project inputs into a highly correlated shared embedding space (Rasiwasia et al. 2010). Recent approaches focus on learning this projection using non-linear neural networks trained through stochastic gradient descent rather than using linear projections found through eigen-decomposition (Jansen et al. 2017). Neural networks can be useful when working with low-level sensory input for which semantic content is not readily accessible through linear transformation alone.

Harwath et al. (2016) presented an architecture, now referred to as "DAVEnet" (Deep Audio Visual Embedding Network), which is trained to learn a semantic embedding space into which images and spoken audio recordings of captions of the images could be mapped. They evaluated their method by looking at the cross-modality retrieval recall scores: e.g., given an image and N audio captions, which one of the N audio captions best describes the image? In addition, Harwath et al. (2016) introduced the captioned Places 205 dataset, often referred to as "Places" in this thesis. The dataset consists of approximately 400,000 images of scenes and associated spoken audio captions, collected via Amazon Mechanical Turk. I describe DAVEnet in further detail in Section 1.3.1.

Sun et al. (2016) showed that image captioning models could be applied to improve automatic speech recognition word error rate when paired images were available for the spoken caption. They used Karpathy and Fei-Fei (2015)'s image captioning model to inform the language model, generating additional textual captions to improve the N-gram language model. Sun et al. (2016) also used the model to rescore the top hypothesis from the ASR beam search using a word-level RNN conditioned on the image. By conditioning on visual context, the authors were able to attain an improvement of 3% absolute word error rate over the baseline audio-only recognizer. This work shows that the semantic information present in an image can reduce uncertainty when parsing speech prompted by the image. The results suggest that there is a large amount of mutual information information shared between images and visually prompted speech.

Harwath and Glass (2017); Harwath et al. (2018a;b) explored whether it might be possible to capitalize on the mutual information shared between images and visually prompted speech to learn key words and phrases for visually salient concepts in an unsupervised manner. Harwath and Glass (2017) conduct a word learning experiment whereby they segment regions of interest in images and audio clips, average pool the embedding regions to a vector, and use k-means on the vectors to find clusters representing concepts. Cluster purity is then evaluated using the words from textual forced-alignments of the spoken captions as labels. The authors find that there are many large, highly pure clusters learned for objects, colors, textures, and scenes.

Harwath et al. (2018b) adds an additional modality to the Places dataset: spoken captions in Hindi. The authors find that using the additional modality during training and learning a similarity metric between the three different pairs of modalities (English speech, Hindi speech, and images) provides additional grounding information during training, improving modality retrieval performance. In addition, the approach enables the unsupervised translation of certain key words and phrased from English to Hindi and Hindi to English.

Harwath et al. (2018a) introduces new methods for calculating the similarity between image and audio embedding maps. In addition, they employ more fine-grained clustering analysis than Harwath and Glass (2017), extracting volumetric regions with high similarity density as the components used for clustering and comparing purities both to the text from the forced-aligned caption and pixel-wise object labels from the ADE20K dataset (Zhou et al. 2016). In addition, they explore the use of component-wise concept detectors for components of the embedding, a similar approach to my word learning experiments in Section 4.6. One difference is that they use the segmentations and labels from ADE20K (Zhou et al. 2016) to choose detectors where there is agreement between image detectors and spoken word detectors. Since (a) the dataset I use in Section 4.6 does not have pixel-wise labels and (b) pixel-wise labels tend to be biased towards objects, I focus solely on word detectors from the audio embeddings. Both Harwath et al. (2018b)'s approach and mine are based on network dissection (Bau et al. 2017; Zhou et al. 2017).

Bau et al. (2017); Zhou et al. (2017) pioneered an approach called "network dissection" in which individual neurons in a fully convolutional architecture are tested for sensitivity to inputs associated with specific labels. The approach is applied to image classification networks using segmented, pixel-wise annotated data as input. The technique involves thresholding the activations of the neuron: only the top 0.5th percentile of activations for the neuron are considered active. Then, the intersect-over-union (IOU) score is calculated between the activation of the neuron and the presence of a label. Detectors with the highest IOU scores are highly label specific. Usually a threshold for IOU is set, such as 0.04, such that only word-component pairs with IOU scores greater than 0.04 are considered detectors. The authors show that when the basis for the hidden state is randomly rotated (preserving the information, but eliminating any "component-wise" alignment), the number of unique detectors decreases, suggesting it may be advantageous during training for individual neurons in the network to learn more interpretable, component-aligned concepts. The word learning experiments I conduct in Section 4.6 are motivated by this finding.

Jansen et al. (2017) use a sampled margin ranking loss similar to Harwath et al. (2016) but with the addition of a "semi-hard-negative" term. The term, explained in detail in Section 1.3.1, involves selecting an impostor example with the greatest similarity still less than the similarity of the ground truth example. The authors use this objective to train a Siamese-style network to learn a similarity function between audio recordings with random jitter applied. We borrow the use of the semi-hard-negative term for use with our multimodal models.

Petridis et al. (2018) introduced a network which takes spoken audio and video as input and predicts the word being spoken. The videos are short 1.16 second clips of lips moving, speaking the word. The network is trained through supervised classification. In contrast, the models I explore are unsupervised and do not use videos of the speaker, but rather videos which the speaker is describing.

Le (2013) uses another approach to unsupervised concept learning: rather than grounding to another modality, the authors take a generative approach. They train an image autoencoder on ten million images and used a classification metric to show the latent space contained information that enabled it to classify faces. Though this could be an avenue for future work, the DAVEnet models I work with in this thesis are not generative and are able to learn high level semantic information solely through grounding to additional modalities.

Monfort et al. (2018) introduce the Moments in Time dataset, a dataset of over 800,000 three second videos labeled with actions taking place in the videos. I give further details about this dataset and use a subset of the dataset augmented with spoken captions in Chapter 4, showing that DAVEnet is capable of learning more action words when trained on the new dataset.

1.3.1 DAVEnet

In this section, I provide a full background on the DAVEnet architecture proposed by Harwath et al. (2016); Harwath and Glass (2017); Harwath et al. (2018a;b). I describe the structure of the model, its training objective, and evaluation procedure. As most of my research focuses on variants of DAVEnet, its training objective, evaluation procedures, and applications of the architecture, giving a full background of the model is important for understanding my contributions.

At a high-level, the goal of DAVEnet is unsupervised speech understanding. Through an unsupervised procedure, DAVEnet is able to learn to recognize semantically relevant features in images and audio captions describing the images.

At training time, the goal of DAVEnet is to learn a similarity function between images and audio captions such that paired images and audio are considered more similar than mismatched (or "impostor") images and audio. More specifically, for a given image, the image's "ground truth" audio caption should be more similar to it than it is to any given impostor audio caption. Likewise, for a given audio caption, the audio caption's ground truth image should be more similar to it than it is to any given impostor image. This goal is realized through the negative sample margin ranking (SMR) loss and variants thereof (Harwath et al. 2016). The objective is $\min_\theta \mathcal{L}_{\text{SMR}}(\mathcal{D};\theta)$, where $\mathcal{L}_{\text{SMR}}$ is defined as:

$$\begin{aligned}\mathcal{L}_{\text{SMR}}(\mathcal{D};\theta) :=\ & \mathbb{E}_{i\sim\mathcal{D}}\Big[\mathbb{E}_{j\sim\mathcal{D}:j\neq i}\left[\max\left(0, m+\text{sim}_\theta(I^{(i)},A^{(j)})-\text{sim}_\theta(I^{(i)},A^{(i)})\right)\right] \\ & \quad + \mathbb{E}_{k\sim\mathcal{D}:k\neq i}\left[\max\left(0, m+\text{sim}_\theta(I^{(k)},A^{(i)})-\text{sim}_\theta(I^{(i)},A^{(i)})\right)\right]\Big]\end{aligned} \tag{1.1}$$

where $i \sim \mathcal{D}$ is an index drawn uniformly from 1 to the size of the dataset, θ are the parameters of the model, m is the margin hyperparameter (typically 1), and $I^{(\cdot)}$ and $A^{(\cdot)}$ are specific images and audio in the dataset, respectively.

In practice, the expectation over i is estimated via minibatch subsampling and batch averaging. Empirically, we have found that approximating the expectations over j and k is best performed with only one negative sample each (two negative samples total). This empirical finding likely indicates that sufficient stochasticity is needed to avoid converging to sub-optimal local optima.

In addition to SMR loss, for some DAVEnet models, the objective function is augmented with a semi-hard-negative loss term proposed in Jansen et al. (2017). The example with the greatest similarity less than the ground-truth similarity is used as the impostor example for margin rank loss. If such an example does not exist, we fall back to uniform negative sampling: sampling an example that differs from the ground truth example uniformly at random. We call this new loss term, (semi-) hard-negative loss ($\mathcal{L}_{\text{HN}}$).

In practice, the two losses are blended together using a hyperparameter, λ_{HN}, typically set to one:

$$\mathcal{L}_{\text{DN}} = \mathcal{L}_{\text{SMR}} + \lambda_{\text{HN}}\mathcal{L}_{\text{HN}} \tag{1.2}$$

DAVEnet is implemented as a neural architecture, shown in Figure 1-2. It consists of two branches learning two functions—f_I and f_A—which perform non-linear transformations on images and audio, mapping them to representations in which simple linear kernel functions can be used to gauge semantic similarity. For example, the

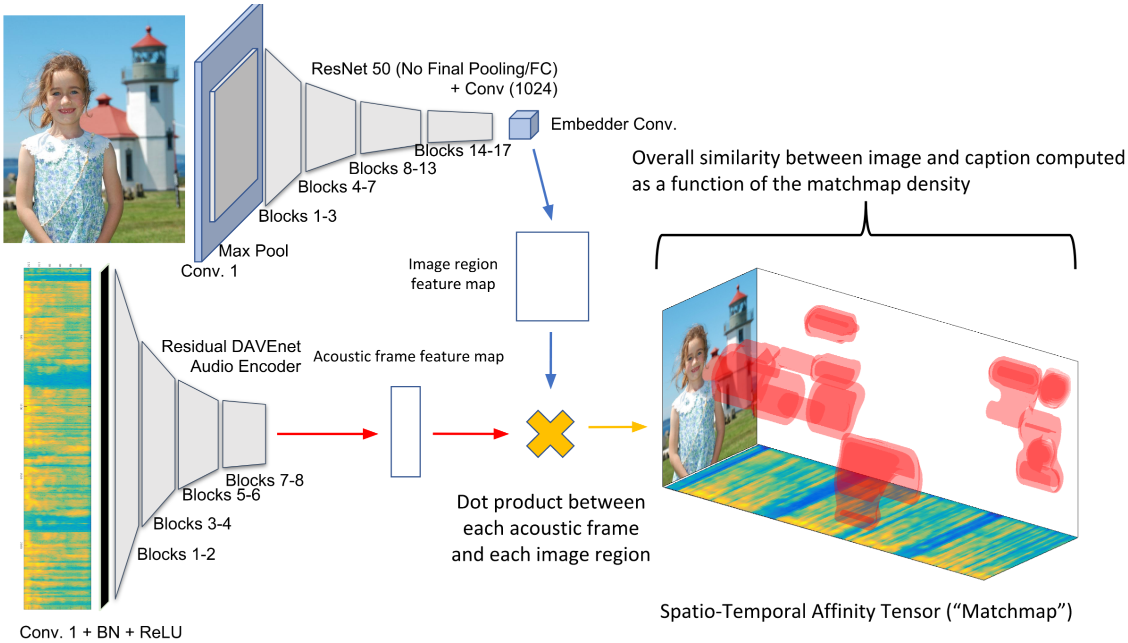

ResNet 50 (No Final Pooling/FC)
+ Conv (1024)
Embedder Conv.
Blocks 14-17
Blocks 8-13
Blocks 4-7
Blocks 1-3
Max Pool
Conv. 1
Overall similarity between image and caption computed as a function of the matchmap density
Image region feature map
Residual DAVEnet Audio Encoder
Acoustic frame feature map
Blocks 7-8
Blocks 5-6
Blocks 3-4
Blocks 1-2
Dot product between each acoustic frame and each image region
Spatio-Temporal Affinity Tensor ("Matchmap")
Conv. 1 + BN + ReLU

Figure 1-2: DAVEnet architecture with residual encoders. Inputs are encoded to embedding maps via modality-specific encoder networks. Similarities are then computed between all pairs of pixels/time steps in the embedding maps to form a matchmap. When embeddings are mean pooled and similarity is computed between vectors, it is mathematically equivalent to averaging the spatio-temporal matchmap similarities but more computationally efficient.

similarity function which we have found to be the best performing empirically is the dot product, though cosine similarity or Euclidean similarity could also be used in theory.

Thus far, the primary evaluation task for DAVEnet has been modality retrieval: given an image, find an audio caption that best describes the image, or given an audio caption, find the most relevant image it could be describing. Recall at 1, 5, and 10 (R@1, R@5, R@10) are used to assess performance at this task.

1.4 My Contributions and Thesis Outline

My contributions are centered around exploring properties of the DAVEnet model, exploring new objectives for training/evaluation-time techniques to change those properties, and applying the architecture to new domains, more specifically, I:

1. Identify and present a solution to the problem of bias towards certain audio captions during modality retrieval,

2. Define the property of modality invariance and propose a regularization term to use during training to encourage the property in the learned embedding space,

3. Apply DAVEnet to a new dataset with additional modalities: the Captioned Moments dataset consisting of three second videos (Monfort et al. 2018), showing that training on this new dataset enables the learning of action-related concepts and concepts grounded to ambient sound.

In Chapter 2, I identify a problem with the current DAVEnet architecture: it tends to have "favorite" audio captions which it chooses to match images far more often than other audio captions. I propose and experiment with ways to downweight this preference for captions that have high similarity a priori.

In Chapter 3, I further explain the property of modality invariance and why a modality invariant embedding space could be desirable. I explain a series of experiments I ran on a smaller digits-based dataset combining MNIST and TIDIGITS

(LeCun et al. 1998; Leonard and Doddington 1993) in which I use a regularization technique borrowed from variational models in a novel way as a means to filter out semantically irrelevant modality information in the learned embedding space. This research was the subject of my ASRU 2017 paper (Leidal et al. 2017).

In Chapter 4, I apply DAVEnet to a new dataset: Captioned Moments, a spoken caption-augmented subset of Moments in Time (Monfort et al. 2018) consisting of short three second videos depicting actions. I discuss the potential challenges and benefits of adding the temporal dimension and ambient audio dimension in videos. Using the technique of network dissection (Bau et al. 2017; Zhou et al. 2017), I analyze the words learned by the model when trained on various datasets, showing that the model tends to learn more action-related concepts and concepts grounded to ambient video audio when fine-tuned on the Captioned Moments dataset.

In Chapter 5, I recap my results and conclude with a discussion regarding the future directions of multimodal learning, language acquisition, and unsupervised learning in general.

Chapter 2

Analysis of DAVEnet

2.1 Introduction

In the current machine learning research climate, it is very common for new model architectures to be introduced in every paper. However, I find it interesting and worthwhile when researchers take an existing model and dissect it: determining what it does well, where it can be improved, and building an intuition for what the model is learning. From there, new models can be designed to address the better understood shortcomings of existing models.

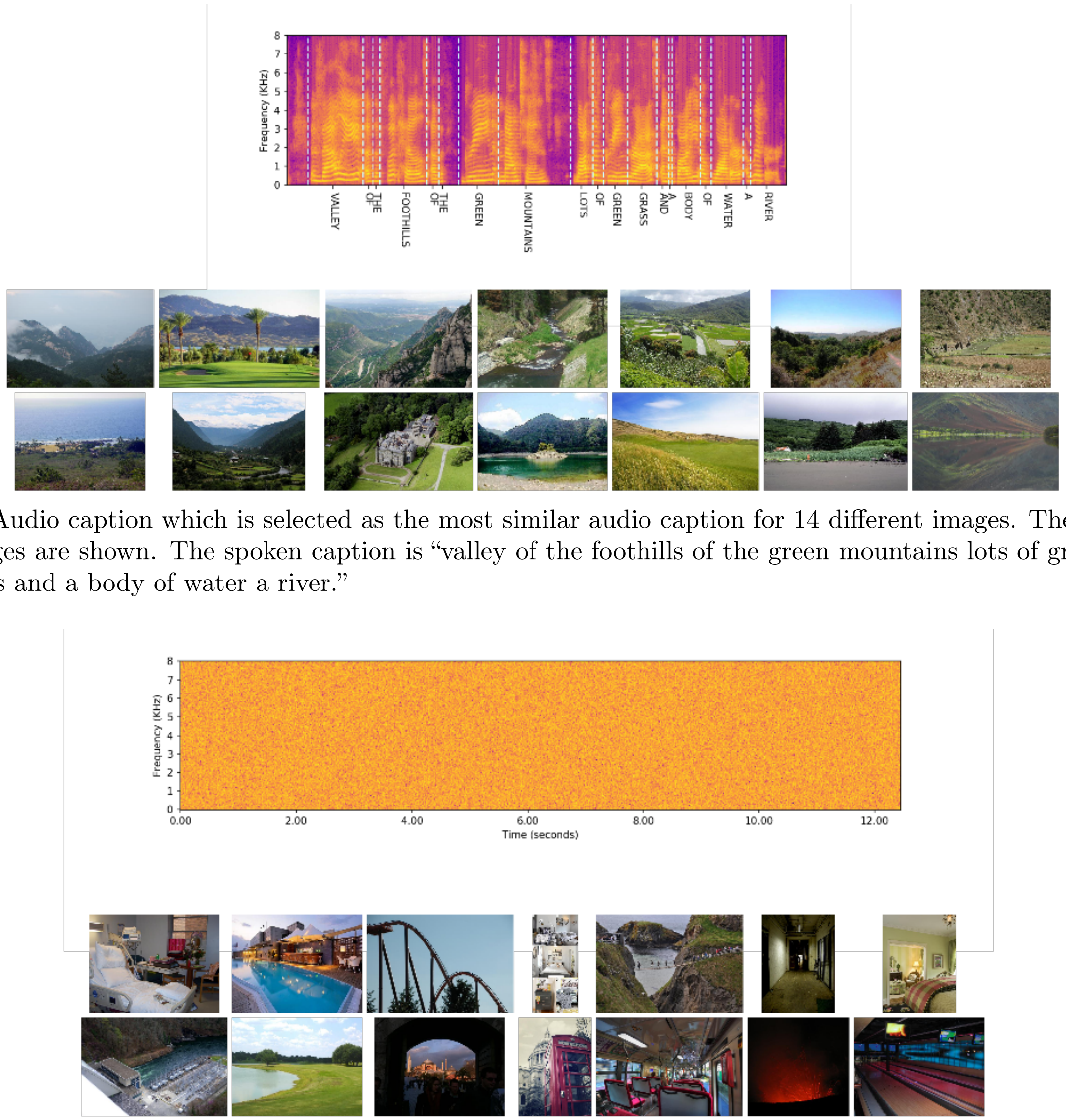
I spent a portion of my time during my thesis analyzing DAVEnet, the topology of its learned embedding space, its properties, and areas for improvement. In this chapter, I give an overview of what I learned and discussions to help build intuition about what DAVEnet might be learning.

I frame the discussion around a specific problem: "favoritism" in modality retrieval. More precisely, I found the model that performs image to caption modality retrieval by choosing the maximum a posteriori caption tends to be biased towards certain audio captions. First, I introduce the problem. Then, I introduce a change I made to the DAVEnet architecture to allow the model to learn two conditional probability mass functions during training: given an image query, what is the probability an audio caption is retrieved for the image, and given an audio caption query, what is the probability an image is retrieved for the audio caption. Finally, I conclude with a discussion regarding how other potential solutions to the bias problem may work. I propose a few ideas for regularization terms/sampling terms to try to combat the problem at training time rather than during evaluation.

All models in this section were trained and evaluated on the Places 205 dataset, referred to simply as "Places" (Harwath et al. 2016; Harwath and Glass 2017; Zhou et al. 2014). The models with image encoders pretrained on ImageNet use a DAVEnet architecture with a Resnet-50 image encoder (He et al. 2016) and residual audio encoder (see Appendix A). The models with non-pretrained image encoders use a DAVEnet architecture with a VGG-16 image encoder (Simonyan and Zisserman 2014) and 5-layer convolutional audio encoder. For both cases, the audio branch has a rectified linear output (constraining its range to the non-negative real subspace: $\{x \in \mathbb{R}^D \mid \forall i : 1 \leq i \leq D.\ x_i \geq 0\}$ where D is the embedding dimension) while the image branch has linear output. All models were trained using the blended hard-negative sampled margin rank objective (Equation 1.2).

2.2 "Favorite" Audio Captions

The primary metric for evaluating DAVEnet is the modality retrieval recall score. Introduced in Section 1.2.2, the task of modality retrieval involves finding the most relevant example in one modality given a query in another modality. In the case of DAVEnet, it involves finding the most similar audio caption to a given image, or the most similar image to a given audio caption, with the goal of matching each query to its ground truth pair. That is, image to caption recall (@1) would be 1.0 if each image is matched to the one caption describing that image. Recall at K (R@K) means the percentage of examples for which the correct ground truth example was in the top K examples when sorted by similarity in decreasing order. The state-of-the-art DAVEnet architecture attains R@10 scores around 70% with a pretrained image encoder and 50% for a non-pretrained image encoder. The exact recall scores for non-pretrained and pretrained models are given in the "Baseline" and "Baseline (P)" rows of Table 2.1. The caption to image direction performs better in both cases, but



(a) Audio caption which is selected as the most similar audio caption for 14 different images. The 14 images are shown. The spoken caption is "valley of the foothills of the green mountains lots of green grass and a body of water a river."

(b) Audio caption containing no speech, only background noise, which is selected as one of the top 10 most similar audio captions for 113 different images. 14 of the 113 images are shown.

Figure 2-1: Examples of "favorite" audio captions for caption retrieval R@1 (a) and R@10 (b)

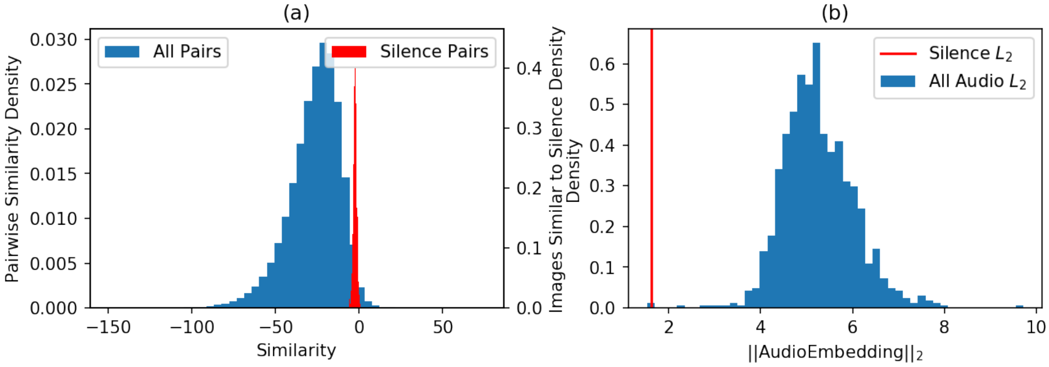


Figure 2-2: (a) The distribution of similarities for paired images with the silent utterance shown in Figure 2-1(b) are shown in red. The distribution of all pairwise similarities is shown in blue. (b) shows the distribution of L_2 norms of the audio embeddings. Note how the silent caption lies close to the origin.

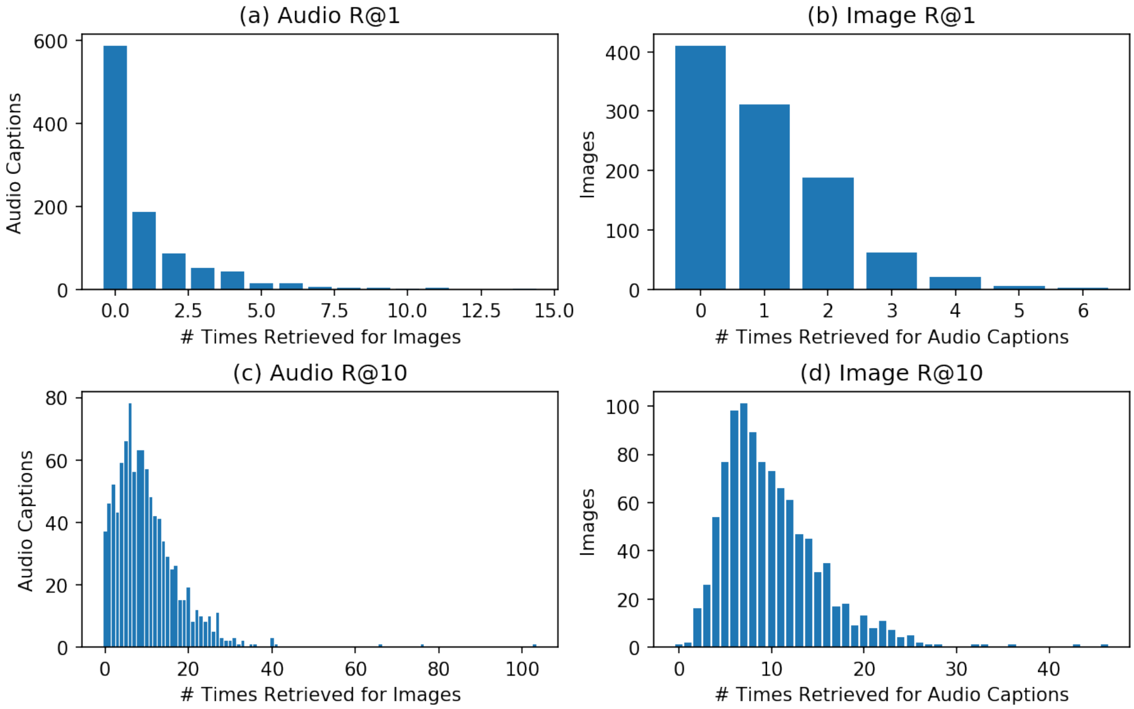


Figure 2-3: (a) and (c) are histograms showing the number of times a given audio caption was in the top-1 or top-10 relevant audio captions for a given image. For example, in (a), the first bar indicates that 586 audio captions were never selected as the most similar audio caption for an image. The last bar in (a) indicates that 1 audio caption was selected as the most similar audio caption for 14 images. (b) and (d) are histograms showing the number of times a given image was in the top-1 or top-10 relevant images for a given audio caption.

especially for the pretrained model.

Curious as to why caption to image modality retrieval tended to outperform image to caption, I looked to more specific statistics about the modality retrieval process. Figure 2-3 shows histograms of the number times each example is selected to be in the top K most similar examples for a query from the other modality for 1000 validation pairs. Figure 2-3(a) shows that 586 audio clips were never selected as the most similar audio clip to an image, yet one audio clip was selected 14 times as the most similar audio clip to an image. That audio clip is shown in Figure 2-1(a).

Though one might argue that the audio clip shown in Figure 2-1(a) is rightfully considered relevant to all 14 images to which it is matched, the task is not truly to retrieve the most similar audio caption, but rather to retrieve the "correct", ground truth audio caption, even if it does not best describe the image. Similarity is merely a heuristic for choosing the correct ground truth audio caption, but it is not always an admissible heuristic. For example, if there is a vague, uninformative caption of an image, it might be considered less similar than a longer, more informative description for a different image. From another perspective, an overly descriptive caption for an image might dilute the more informative content, leading it to be considered less similar than a more concise description. Finally, since image embeddings can be negative, there is a notion of "dissimilarity": if an image does not align with the content of a caption, the pair may have a negative similarity. If the caption contains few if any recognized words, the caption may nonetheless have high relative similarity to images, as there is no recognizable content to prove dissimilarity. Though this last case might seem construed, there is evidence suggesting it may be the case: the audio clip which was found to be in the top ten most similar audio clips for 113 different images contains no speech, only background noise. This silent audio caption has pairwise similarity scores with images that are very close to zero (well above the mean of -26.7) and its embedding lies much closer to the origin than average (see Figure 2-2), suggesting the audio embedding may zero-out negative components of the image embedding during the dot product.

Regardless of the properties of the similarity metric and its admissibility as a heuristic for modality retrieval, the fact that 586 audio clips were never selected to match an image means that only 414 unique audio clips were selected at least once to match an image, setting the image to caption R@1 upperbound at 41.4%. In the context of post-processing, this upper-bound poses a barrier for improving recall based on the scores derived from an already-trained model. In addition, one can view the silent audio caption which occurs in the top ten captions for 113 images as a "wasted" slot in the top ten for each of those images. With this in mind, I set out to find a solution to the highly skewed caption recall distribution.

2.3 My Solution: Compensating for the Prior

Figure 2-3 shows that some audio captions are favored over others a priori. To study this further, I transform similarity scores into conditional probability distributions using the softmax function. From this point of view, the current procedure for image to caption modality retrieval involves selecting the maximum a posteriori caption given and image. For a set of N images and M captions, I now show how to convert the similarity score matrix, S, into conditional probability distributions and approximate prior distributions. First I introduce notation.

Suppose instead of the process of modality retrieval being a deterministic process, it is a stochastic process where the retrieved image given a query is selected randomly according to a distribution. Since there are finite images and captions, we can assign each an ID[1]. Let Q_I be the random variable representing the ID of the queried image for image to caption retrieval. Let R_A be the random variable representing the ID of the retrieved caption for image to caption retrieval. Let Q_A be the random variable representing the ID of the queried caption for caption to image retrieval. Let R_I be the random variable representing the ID of the retrieved image for caption to image retrieval. Let I be the N by D matrix of image embeddings (where D is the embedding dimension) and A be the M by D matrix of audio embeddings. S

[1] In practice, it is most convenient if the IDs of paired audio captions are equal, representing the diagonal of the similarity matrix. However, the theory holds without this assumption.

represents the similarity matrix.

The goal is to compensate for the skewed distribution $P(R_A \mid Q_I)$, but first similarities must be converted into probabilities. I do so with the softmax function with temperature hyperparameter τ.

$$
\begin{aligned}
S &= \Big[\mathrm{sim}(I_i, A_j)\Big]_{i,j} \\
P(R_I = i | Q_A = a) &= \frac{e^{\frac{S_{i,a}}{\tau}}}{\sum_{i'=1}^{N} e^{\frac{S_{i',a}}{\tau}}} \\
P(R_A = a | Q_I = i) &= \frac{e^{\frac{S_{i,a}}{\tau}}}{\sum_{a'=1}^{M} e^{\frac{S_{i,a'}}{\tau}}}
\end{aligned}
$$

In the case of dot product similarity, $S = IA^T$. For numeric stability, the max of the similarities along the axis being softmaxed is often subtracted before exponentiation.

The prior of a given audio caption can then be estimated as:

$$
\begin{aligned}
P(R_A = a) &= \mathbb{E}_{i \sim P(Q_I)}\left[P(R_A = a \mid Q_I = i)\right] \\
&= \sum_{i=1}^{N} P(R_A = a \mid Q_I = i) P(Q_I = i) \\
&= \frac{1}{N} \sum_{i=1}^{N} P(R_A = a \mid Q_I = i)
\end{aligned}
$$

assuming each of the N images is equally likely to be queried.

As one would expect, for a given caption, a, the $P(R_A = a)$ is highly correlated to the number of images for which a is the most similar audio caption ($|\{i | \forall i : 1 \leq i \leq N \wedge a = \mathrm{argmax}_{a'} S_{i,a'}\}|$). This correlation is shown in Figure 2-4(a) ($R^2 = 0.971$).

To compensate for the model's tendency to select the same audio captions repeatedly, we can divide the posterior, $P(R_A \mid Q_I)$, by the prior, $P(R_A)$, to attain an odds ratio which we call the posterior-inverse-prior (PIP), or posterior-prior^{-1}:

$$
\mathrm{PIP}(a \mid i) = \frac{P(R_A = a \mid Q_I = i)}{P(R_A = a)} = \frac{P(Q_I = i \mid R_A = a)}{P(Q_I = i)} \propto P(Q_I = i \mid R_A = a) \tag{2.1}
$$

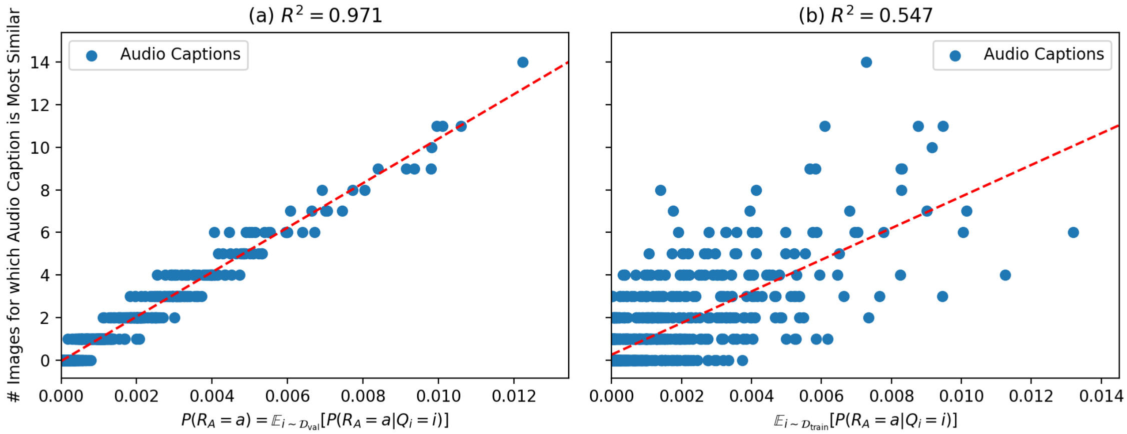


Figure 2-4: Correlation between number of images for which a given audio caption is the most similar audio caption to the image and $P(R_A = a) = \mathbb{E}_{i\sim\mathcal{D}}[P(R_A = a|Q_I = i)]$, with the expectation estimated over images in (a) the validation set and (b) the training set. $\tau = 1.0$ was used for the softmax.

Odds ratios are common-place in information theory. For example, the expected log odds ratio between the posterior and prior over a dataset is the conditional entropy, representing the expected number of bits of information the condition provides (Cover and Thomas 1991). Intuitively, the odds ratio represents a notion of "surprise", or more specifically a quantifier of the change in belief, called information. For example, if Alice asks Bob which car will win in a race: a NASCAR race car or a Toyota Camry, Bob would almost definitely say the race car. If, however, Bob was then informed that the race car had flat tires, Bob's belief that the Camry would win would increase drastically. The information that the race car's tires were flat changed Bob's belief drastically and therefore would have a high posterior prior ratio. On the other hand, if Bob were instead have been told the Camry had flat tires, his belief would not change much, as he already thought that the race car would win. Being informed that the Camry had flat tires therefore has a posterior prior ratio close to one. The ratio can be less than one if belief decreases.

The traditional image to caption modality retrieval process is: for a given i, select the a which maximizes $S_{i,a}$. I now change that to: for a given i, select the a which maximizes $\text{PIP}(a \mid i)$. Intuitively, this can be thought of as choosing the audio caption which seems significantly more likely after seeing the image than before. The greater the change in probability (the greater the "surprise") is, the higher the score is.

There is one caveat: since $P(R_A = a)$ was estimated via an expectation over images, the task has been changed. Originally, the task was: given one image and M captions, find the caption most similar to the image. With the change, the task becomes: given N images—one of which is the query—and M captions, find the caption most similar to the queried image. In other words, using PIP as a direct substitution for similarity provides more information as input than in the original evaluation procedure. However, the use of this additional information can be avoided if $P(R_A = a)$ is approximated via an expectation over images in the training set rather than the validation set. These images in the training set can be considered memorized parameters of the model, and therefore PIP serves as a fair comparison to using similarity directly. Though the $P(R_A = a)$ estimated via an expectation over the training images[2] is not as correlated to the number of images to which the audio caption was most similar as when estimated via an expectation over validation images (see Figure 2-4(b) versus Figure 2-4(a)), there is still a correlation ($R^2 = 0.547$).

2.4 Results

Evaluating PIP($a \mid i$) and PIP($i \mid a$) (calculated in the same manner but with priors over images estimated from training captions) for modality retrieval recall @1, @5, and @10 yields the results given in Table 2.1 in the Posterior-Prior^{-1} rows. The recall statistics in italics do not serve as a fair point of comparison, since they were collected using validation examples to approximate the prior. This caveat is described in further detail at the end of Section 2.3. However, the italicized metrics act as oracle recalls for the ideal case where the expectation of the posterior over queried training examples equals the expectation over queried validation examples.

First I will discuss the results for image to caption retrieval. For both the pretrained model, the improvement was greatest for R@1 with 21.9% relative improve-

[2] a Monte Carlo estimate with 1000 images from the training set was used rather than the full 402,385 images available in the training set.

ment as opposed to only 6.7% and 4.2% for R@5 and R@10, respectively. This suggest that the posterior-inverse-prior technique helps best at sorting out the order among the top few retrieved examples but has less effect on the total sorting. For the non-pretrained model, the effect of the technique is less pronounced for R@1 than for the pretrained case, but more pronounced for R@5 and R@10: 17.5%, 17.9%, 7.9% relative improvement, respectively. The improvement is visually apparent in Figure 2-5(b), as the posterior-prior^{-1} generally has more images with low-index ground truth captions than the baseline.

The results for caption to image retrieval are more mixed. There are cases were the baseline outperforms posterior-prior^{-1} in Table 2.1, and in the cases where that is not the case, posterior-prior^{-1} only attains slight improvements. This is visually evident in the similarity of the baseline and posterior-prior^{-1} curves in Figure 2-5(a).

The fact that recall improved is a good result of using the posterior-prior^{-1} score for modality retrieval, but it does not show whether or not the original problem of the model's bias towards certain audio captions was solved. A histogram of the number of times audio captions were retrieved for a given image (shown in Figure 2-6(a) and Figure 2-6(b)) answers that question.

For image to caption R@1, 449 audio captions were never selected as the most similar audio caption to an image, reduced from 586 using similarity alone. The maximum number of times an audio caption was selected as the most similar caption

	Caption to Image			Image to Caption		
Model	R@1	R@5	R@10	R@1	R@5	R@10
Baseline	0.145	0.386	**0.499**	0.120	0.335	0.468
Posterior-Prior^{-1} ($\tau = 6.2$)	**0.150**	**0.401**	0.499	**0.141**	**0.395**	**0.505**
(*) Posterior-Prior^{-1} ($\tau = 6.2$)	*0.155*	*0.416*	*0.514*	*0.147*	*0.401*	*0.512*
Baseline (P)	0.273	**0.606**	**0.735**	0.219	0.564	0.687
Posterior-Prior^{-1} ($\tau = 3.0$) (P)	**0.292**	0.604	0.730	**0.267**	**0.602**	**0.716**
(*) Posterior-Prior^{-1} ($\tau = 3.0$) (P)	*0.301*	*0.622*	*0.743*	*0.277*	*0.623*	*0.733*

Table 2.1: Results of prior-compensation experiments as compared to the baseline. (P) indicates the image encoder was pretrained on ImageNet. (*) indicates that the prior was calculated using examples of the queried modality from the validation set rather than the training set (changing the task).

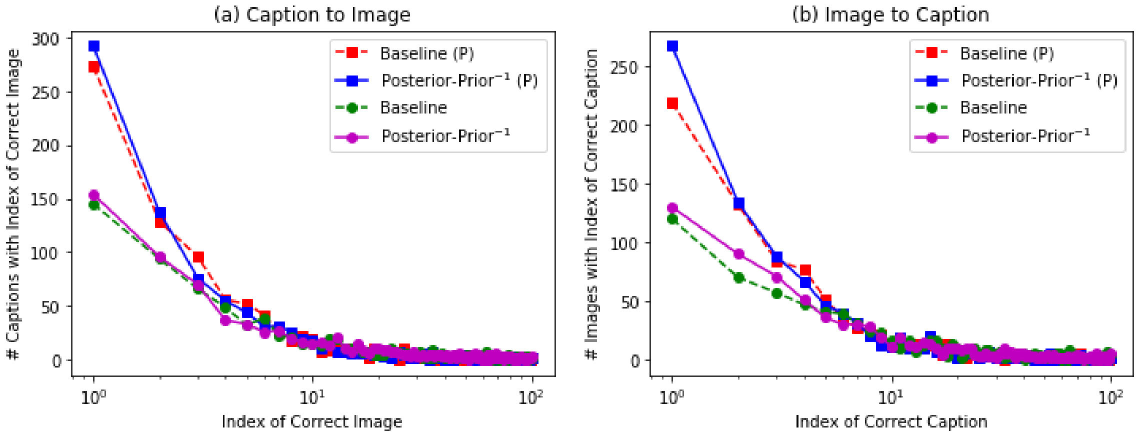


Figure 2-5: A histogram of the index of the ground truth "correct" example in the list of retrieved examples sorted in descending order by various similarity metrics. (P) indicates that the model used was pretrained. For example, the point in the upper left of (b) indicates that 267 images for a pretrained model using Posterior-Prior^{-1} had the correct ground truth audio caption as the most similar audio caption (corresponding to 0.267 R@1 reported in Table 2.1).

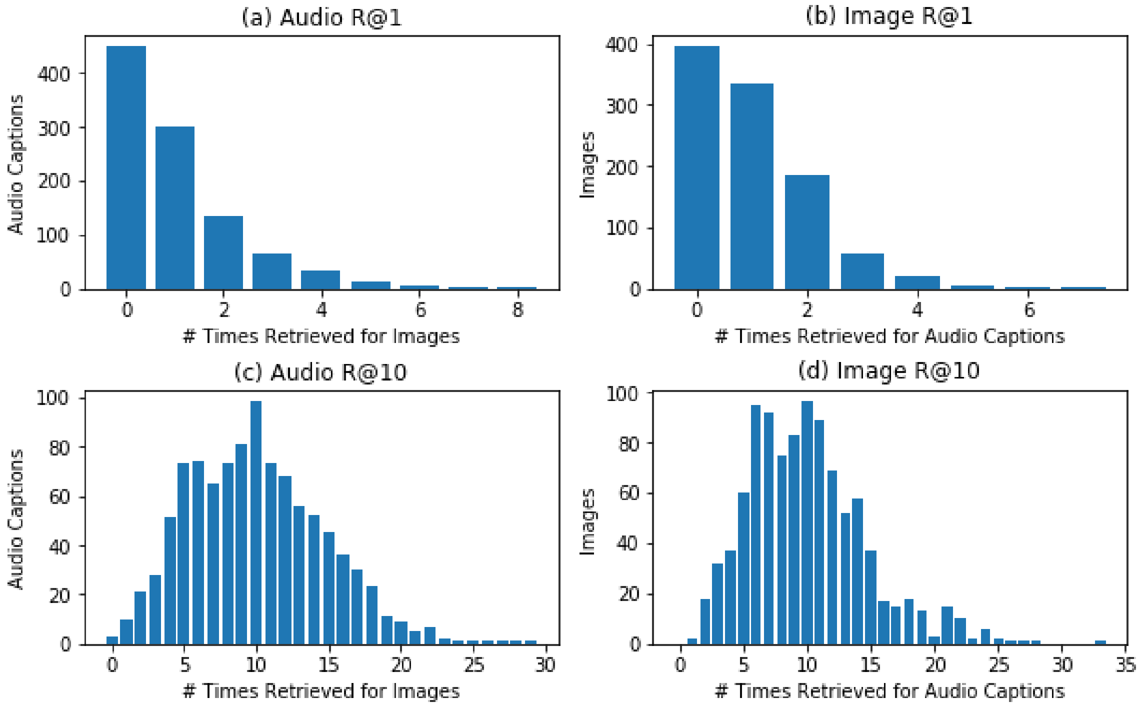


Figure 2-6: A histogram identical to Figure 2-3, except using the posterior-priorsuperscript score for modality retrieval rather than the similarity score.

for an image was 8, down from 14.

For image to caption R@10, three audio captions were never selected in the top ten audio captions for an image, reduced from 37 using similarity alone. The maximum number of times an audio caption was selected in the top ten audio captions for an image was 29, down from 113.

These results suggest that the posterior-prior^{-1} score is an effective evaluation-time technique for reducing bias during modality retrieval tasks.

2.5 Discussion: Other Potential Solutions

One might be tempted to try to use the posterior-prior^{-1} technique during training rather than only during evaluation. In this procedure, one would perform a softmax on the mini-batch similarity matrix, then marginalize to approximate $P(R_A)$ and $P(R_I)$, and then calculate the batch's log-odds ratios $\mathrm{PIP}(i \mid a)$ and $\mathrm{PIP}(a \mid i)$. Sample margin ranked loss with hard negative blending (as given in Equation 1.2) could then be used on the log-odds ratios as opposed to the similarity matrix directly, as in the original formulation. Preliminary experiments with this architecture empirically showed that this training objective was not stable and quickly led to infinite gradients.

I did, however, find that simply performing a softmax on the similarity matrix and performed sampled margin rank loss with hard negative blending on $P(R_A \mid Q_I)$ and $P(R_I \mid Q_A)$ rather than on S led to results equivalent to, if not slightly better than, the baseline. In this setup, I found it beneficial to use a global parameter, τ, the learned temperature for the softmax. This could be beneficial for future variants of the model, as the model learns using a probability mass function over examples, which can be useful for regularization, negative sampling, and other techniques which depend on a probability mass function. For example, during sampled margin rank loss, instead of negative sampling uniformly, one might sample a categorical distribution over examples with weights proportional to the posterior over examples in the minibatch. Another option might be to use the probability mass function to regularize the prior. For example, one could use the Kullback-Liebler (KL) divergence, also known as relative entropy, of the empirical prior from the uniform distribution as a regularization term:

$$\begin{aligned}\mathcal{L} = \mathcal{L}_{\text{DN}} + \lambda_{\text{KL}} \Big(& D_{\text{KL}} \left(\mathbb{E}_a \left[P(R_I \mid Q_A = a) \right] \; || \; \text{Uniform}(N) \right) \\ & + D_{\text{KL}} \left(\mathbb{E}_i \left[P(R_A \mid Q_I = i) \right] \; || \; \text{Uniform}(N) \right) \Big)\end{aligned}$$

which adds a loss term which increases as the empirical prior deviates from the uniform distribution.

Another avenue which should be explored is explicitly training the model on silence and white noise, both for audio and images. Such inputs should not be similar to any non-silent input. This could help counteract the problem shown in Figure 2-1b where the silent audio was similar to many images. In addition, jitter in pixel values should be added to inputs at training time to avoid memorization.

2.6 Conclusion

In this chapter, I looked in detail at the behavior of a DAVEnet architecture in the context of a modality retrieval task. In particular, I looked at image to caption modality retrieval, identifying a problem where the model was biased to certain audio captions. I proposed and evaluated a solution to that problem involving compensating for the prior at evaluation time. Finally, I proposed potential changes to the model during training time, including new sampling procedures and regularization terms, that may help combat the problem during training for future work on the model.

Chapter 3

Modality Invariance

3.1 Introduction

Humans perceive the world through an array of specialized sensory organs: the eyes, ears, nose, tongue, and skin. By processing the stream of signals from these organs, we are able to perceive and understand the world around us. Each of these organs provides information about our surroundings from a different modality: sight, sound, smell, taste, and touch, respectively. By the same token, machine learning models can be given input data from various sensory channels, or modalities, as well.

A modality invariant representation of data is a transformed representation of sensory information such that it is impossible to predict the original modality from the transformed representation. More specifically:

Definition 3.1.1. *Modality invariance.* Given two modalities, $\mathcal{I}$ and $\mathcal{A}$, an embedding space $\mathcal{S}$, and two functions:

$$f_I : \mathcal{I} \mapsto \mathcal{S}$$

$$f_A : \mathcal{A} \mapsto \mathcal{S}$$

The pair $< f_I, f_A >$ is considered modality invariant if it is impossible to learn a function $\mathcal{S} \mapsto \{A, I\}$ which classifies embeddings as their original modality with

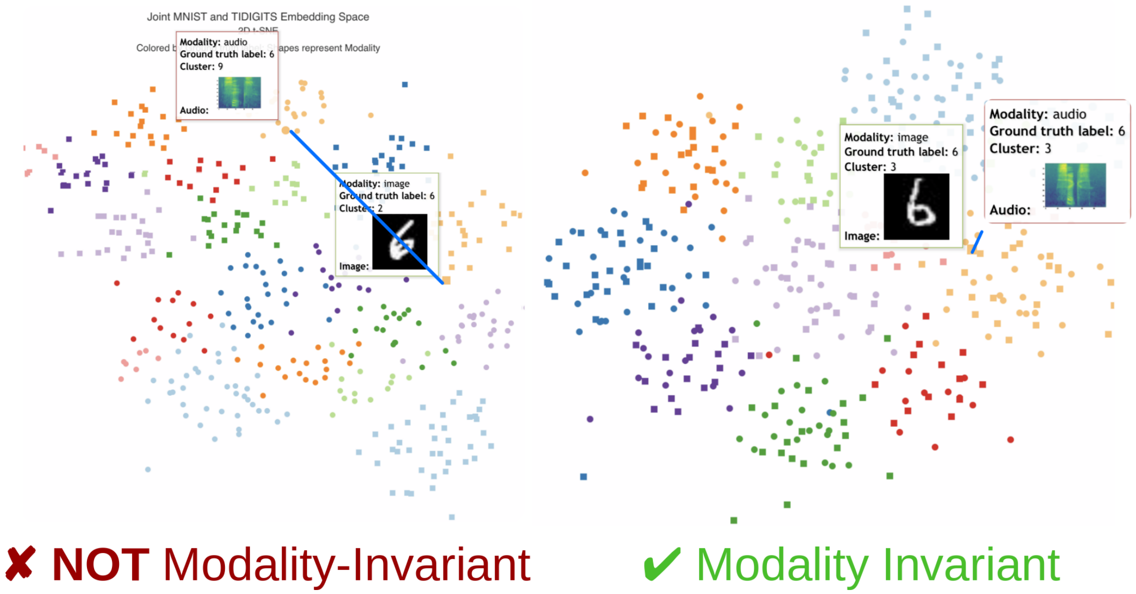


Figure 3-1: A non-modality invariant embedding space (left) versus a modality invariant embedding space (right). The embedding space is a 2D t-SNE projection of the 128 dimensional embedding space learned by the model introduce in Section 3.3. It contains images of handwritten digits (squares) and spoken audio recordings of digits (circles). The color of the point corresponds to the ground truth digit label.

probability sufficiently greater than chance.

For an example of a non-modality invariant embedding space versus a modality invariant embedding space, see Figure 3-1. These embedding spaces are produced from a model I describe in Chapter 3. In the left, there are clusters that correspond to digit-modality pairs, whereas in the right, the clusters only correspond to digits. Modality information has been filtered out when learning the encoder for the second embedding space.

In the trivial case, the transformation could be an injection to some constant, like zero, or in general sampling from any distribution independent of the modality. In that case, the representation is modality invariant, but also contains no semantic information present in original signal. Another example of a modality invariant representation is to learn a classification function on the separate modalities in a supervised fashion. So long as the empirically observed prior distributions[1] over predicted labels

[1] the expectation of the learned posterior over all conditions.

for the classifiers for both modalities are the same, the predicted labels can be considered a modality invariant representation, albeit learned in a supervised manner. The techniques in this chapter will explore unsupervised techniques for learning modality invariant, semantically rich representations.

3.1.1 Benefits of Modality Invariance

What are the benefits of modality invariance? For one, learning a function on the embedding space becomes easier. In other words, filtering out modality information can be viewed as a way to de-noise more important semantic information for tasks which only pertain to the semantic content expressed in the modalities rather than the form of expression itself. An example of such a task is digit classification. For instance, imagine spoken and handwritten digits between zero and nine, inclusive, are encoded to a shared embedding space. If the embedding space is modality invariant, digits will occupy the same subspace, regardless of whether they were spoken or handwritten. A much weaker classifier could then be used to classify the embeddings. This can be expressed more formally by showing that modality invariant embeddings can be classified more accurately by classification models with less parameters than non-modality invariant embeddings.

The function on the embedding space does not have to be a discriminative classifier. It could also generate output in another modality. For example, it could map the embedding to generated spoken audio or a generated image. This could be useful for modality translation, because only one decoder function per decoded modality needs to be learned if there is a common, modality invariant input representation. To make an analogy to another area of computer science: it is similar to the LLVM compiler, which has multiple frontend syntaxes and multiple backend architectures but one shared intermediate representation: there only needs to be a function mapping the intermediate representation to the machine code for a specific architecture. The benefit of the shared representation lies in the number of functions required. Imagine there are four input modalities and four output modalities. To be able to convert between any pair of input and output modalities directly, one would have to learn

$4! = 24$ functions. But, if there is a shared intermediate representation, only $4+4=8$ functions are required (one for each input modality to intermediate representation and one from the intermediate representation to each output modality).

In addition to the practical benefits of a modality invariant embedding space, the goal is biologically inspired as well. More specifically, psychological studies show children are able to learn through associating stimuli during their early years. For example, a child hearing his or her mother pronounce “seven” or write a “7” might learn to think of the same concept upon hearing or seeing either. In fact, Man et al. (2012) showed that the temporoparietal cortex of the human brain produces content-specific and modality-invariant neural responses to audio and visual stimuli.

3.1.2 My Approach

In general, when learning a latent space for unsupervised concept discovery[2], it is advantageous to filter out an information considered noise unrelated to the semantic concepts of interest. For example, if the goal is to train a model which can learn to distinguish and identify sensory inputs representing digits in the same way a human would, it is important for the model to filter out sensory noise that does not affect the underlying semantic content. To this end, in this chapter I take a closer look at a technique for regularizing the learning process using a conditional entropy loss term to filter out undesired information.

I train a neural model based on DAVEnet to map speech audio and image inputs into a modality invariant semantic embedding space. In my method, I map image and audio inputs to the parameterizations of diagonal Gaussians representing the posterior distribution over semantic embeddings. I then sample embeddings from this distribution and use sampled margin rank loss (defined in Equation 1.1) to encourage samples from paired audio and image inputs to be more similar than mismatched pairs of audio and images. Although (Harwath et al. 2018a; 2016) have shown DAVEnet can learn a semantically rich embedding space with this objective, the embedding space learned by DAVEnet with sampled-margin-rank loss alone is not modality invariant.

[2] by clustering points in the latent space, for instance

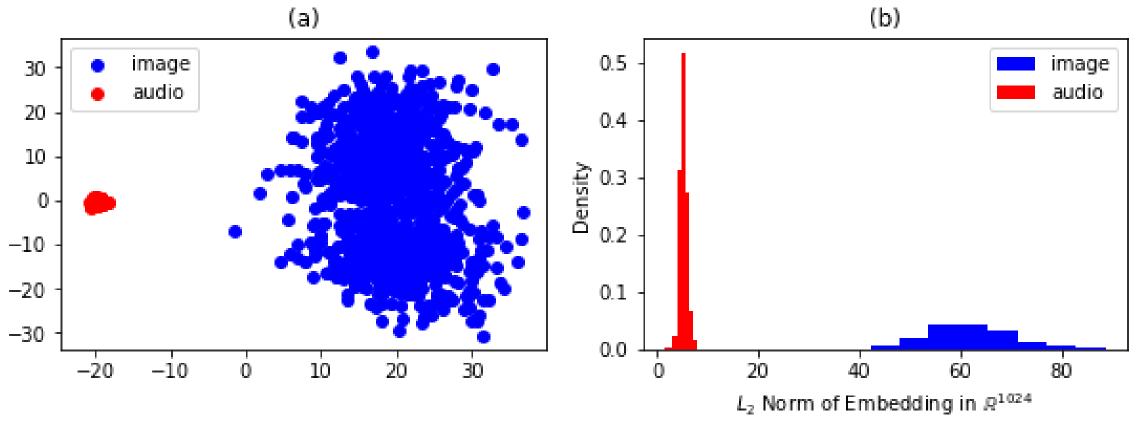


Figure 3-2: (a) 2D PCA of 1024-dimensional embedding space and (b) L_2 norm of 1024 dimension embeddings for Imagenet-Pretrained Resnet-DAVEnet trained on Places

For example, Figure 3-2 shows that image and audio captions occupy separate subspaces of the learned embedding space when DAVEnet is trained on Places.

In my approach, I explore methods of better encouraging modality-invariance. That is, not only should semantically similar content within the same modality be clustered in the embedding space, but the distributions of embeddings for semantically equivalent audio and images should be the same. This goal is based on the assumption that modality-specific information is effectively noise for tasks requiring only the semantic content of the sensory input.

My experiments use a simpler dataset and architecture than DAVEnet. I focus on a combined dataset consisting of pairs of spoken and handwritten digits ranging from 0 to 9. The spoken digits are drawn from the TIDIGITS corpus (Leonard and Doddington 1993) while the handwritten digits are from MNIST (LeCun et al. 1998).

To drive the posterior distributions over embeddings to be the same for semantically equivalent inputs across modalities, I introduce a term to the objective which regularizes the amount of information encoded in the semantic embedding. The term, borrowed from variational autoencoders (VAEs), is the KL divergence of the posterior distribution from the unit Gaussian. My results suggest that when this regularization term is increased from zero during hyperparameter tuning, modality-information tends to be filtered out prior to semantic-information. I believe this regularization technique has the potential to be useful for filtering out information in information-rich embedding spaces in general.

3.2 Previous Work

Saito et al. (2016) developed an adversarial neural architecture to learn modality-invariant representations of paired images and text. Modality-invariance was encouraged using an adversarial setup in which the discriminator was given one of the two representations or a sample drawn from the unit Gaussian. The discriminator was tasked with determining which modality the input originated from or whether it was drawn from the unit Gaussian. The encoders were trained through gradient reversal, as used previously in adversarial domain adaptation and generative adversarial networks (Saito et al. 2016; Ganin and Lempitsky 2015; Ganin et al. 2016; Tzeng et al. 2017; Goodfellow et al. 2014).

Kashyap (2017) also applied Harwath et al. (2016)'s approach to the MNIST and TIDIGITs dataset, focusing primarily on using the embeddings for cross-modality transfer learning. My early work with MNIST and TIDIGITS focuses more on the learned embedding space itself, and methods to promote modality-invariance.

Hsu et al. (2017a) designed a convolutional variational autoencoder (CVAE) for log Mel-filterbanks of speech drawn from the TIMIT dataset. In my work concerning MNIST and TIDIGITS, I use the same convolutional network architecture for my audio encoder network.

For the multi-modality MNIST-TIDIGITS work conducted so far, the network architecture and loss function is based on Harwath et al. (2016), but instead of deterministically mapping inputs to embeddings, I map inputs to the parameterization of a diagonal Gaussian, and sample embeddings from it. In addition, I add a regularization term for the posterior distributions. In this regard, my method takes a similar approach to achieving modality-invariance as Saito et al. (2016) insofar as we both drive the distribution of embeddings to have minimal deviation from a unit Gaussian prior distribution of embeddings. I have also empirically found that at least in cases where semantic information is discrete and closed, encoders can deceive a discriminator without using gradient reversal. In addition, the problem of modality-invariant embeddings using speech as one of the modalities has yet to be explored, so my research makes a novel contribution in this area.

3.3 Methods

I will first formalize the problem. Given a set of co-occurring images and captions, $(x_v^{(i)}, x_a^{(i)}), i = 1...N$ where $x_v^{(i)} \in \mathcal{V}$ (image space) and $x_a^{(i)} \in \mathcal{A}$ (audio caption space), functions $f_v \in \mathcal{F}_v : \mathcal{V} \mapsto \mathbb{R}^D$ and $f_a \in \mathcal{F}_a : \mathcal{A} \mapsto \mathbb{R}^D$ are chosen to optimize some objective that promotes the encoding of semantic information contained in the inputs $x_v^{(i)}$ and $x_a^{(i)}$ into $f_v(x_v^{(i)})$ and $f_a(x_a^{(i)})$, respectively. D is the latent dimension. For example, if $x_v^{(i)}$ is a picture of a handwritten "7" and $x_a^{(i)}$ is an audio recording of someone saying "seven", $f_v(x_v^{(i)})$ and $f_a(x_a^{(i)})$ should be considered highly semantically related by some similarity metric. As with DAVEnet, I aim to increase the margin between the similarity of representations of co-occurring inputs and the similarity of representations of non-co-occurring inputs using sampled margin rank loss, defined in Equation 1.1. Note that hard-negative margin rank loss is not used for this model. For brevity, we refer to this loss as *similarity loss*, $\mathcal{L}_{\text{sim}}$.

In contrast to DAVEnet, my encoders, f_v and f_a, are non-deterministic. The model learns the deterministic functions $\mu_v : \mathcal{V} \mapsto \mathbb{R}^D$ and $\log \sigma_v^2 : \mathcal{V} \mapsto \mathbb{R}^D$. Then $\mu_v(x_v^{(i)})$ and $\log \sigma_v^2(x_v^{(i)})$ are used to parameterize a diagonal Gaussian representing

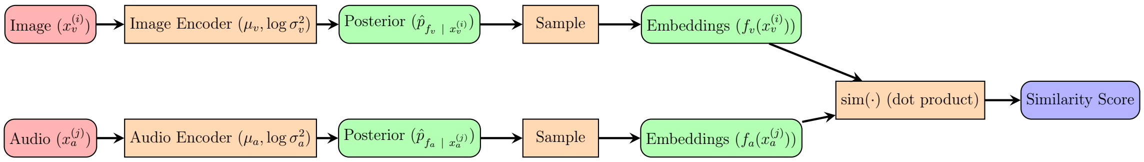


Image ($x_v^{(i)}$)
Image Encoder ($\mu_v, \log \sigma_v^2$)
Posterior ($\hat{p}_{f_v \mid x_v^{(i)}}$)
Sample
Embeddings ($f_v(x_v^{(i)})$)
Audio ($x_a^{(j)}$)
Audio Encoder ($\mu_a, \log \sigma_a^2$)
Posterior ($\hat{p}_{f_a \mid x_a^{(j)}}$)
Sample
Embeddings ($f_a(x_a^{(j)})$)
sim($\cdot$) (dot product)
Similarity Score

Figure 3-3: The high-level model structure. The image/audio are encoded to the parameterization of the posterior distribution. Embeddings are then sampled from the parameterized posterior distributions and used to calculated the similarity using the dot product.

the posterior distribution over embeddings:

$$\hat{p}_{f_v \mid x_v^{(i)}} := \mathcal{N}\left(\mu_v(x_v^{(i)}), \operatorname{diag}\left(\sigma_v^2(x_v^{(i)})\right)\right) \tag{3.1}$$

Embeddings are then sampled from the posterior:

$$f_v(x_v^{(i)}) \sim \hat{p}_{f_v \mid x_v^{(i)}}$$

and likewise for f_a. This process is illustrated in Figure 3-3. During training, 16 samples were sampled per input point and the re-parameterization trick described in Kingma and Welling (2013) was used to backpropagate through the sampling process.

In addition to $\mathcal{L}_{\text{sim}}$, I used the KL divergence of the predicted posteriors (diagonal Gaussians) over embeddings from the prior over embeddings (the unit Gaussian) as a regularization term I call *information gain (IG) loss*:

$$\begin{aligned}\mathcal{L}_{\text{IG}} = \mathbb{E}_{i \sim \mathcal{D}}\big[&KL(\hat{p}_{f_v \mid x_v^{(i)}} \mid\mid \mathcal{N}(0, I_z)) \\ &+KL(\hat{p}_{f_a \mid x_a^{(i)}} \mid\mid \mathcal{N}(0, I_z))\big]\end{aligned}$$

In practice, the expectation is approximated over the minibatch.

The total loss function is then:

$$\mathcal{L} = \mathcal{L}_{\text{Sim}} + \lambda_{\text{IG}}\mathcal{L}_{\text{IG}} + \lambda_{\text{WD}}\mathcal{L}_{\text{WD}} \tag{3.2}$$

where $\mathcal{L}_{\text{WD}}$ is the sum of all Frobenius norms of weight matrices and convolutional kernels, and λ_{IG} and λ_{WD} are tunable hyperparameters.

3.4 Datasets

For images, I used the MNIST dataset of handwritten digits (LeCun et al. 1998). The dataset contains 60K training images and 10K test images. The images are 28x28 8-bit grayscale images, and each is preprocessed to have pixel values between 0 and 1. For audio, I use the TIDIGITS dataset of spoken utterances sampled at 20 KHz

(Leonard and Doddington 1993). I only use digit strings containing a single digit from men, women, and children. After filtering out utterances which contain more than one digit, 6,456 training utterances, 1,076 test utterances, and 1,076 validation utterances remain. Using the Kaldi speech recognition toolkit (Povey et al. 2011), 80 dimensional log Mel-filterbank features were calculated with a 25ms window size and a 10ms frame shift, using a Povey window[3]. To create inputs of the same size, I pad or crop each spectrogram to 100 frames (i.e., one second of speech) which is one frame longer than the mean frame length of the available utterances. I preprocessed each filterbank to have zero mean and unit variance. Longer utterances were center cropped. Shorter utterances were zero padded at the end after adjusting the filterbank to have zero mean. For TIDIGITS, I also combined the utterances labeled "oh" and "zero" into one class for the purpose of labeling clusters during analysis[4].

3.5 Experiments

I used convolutional neural networks to predict the parameterizations of $\hat{p}_{f_v \mid x_v^{(i)}}$ and $\hat{p}_{f_a \mid x_a^{(i)}}$ (Equation 3.1). I trained the networks to minimize Equation 3.2 for the MNIST and TIDIGITS datasets described in Section 3.4. I compared the embedding spaces produced when $\lambda_{\text{IG}} = 0$ and when $\lambda_{\text{IG}} > 0$ to gauge the effect of regularizing information gain in the posterior.

I set the embedding dimension to be $D = 128$, which is consistent with the latent embedding dimensionality used by Hsu et al. (2017a) for their variational autoencoder for 58 phones. I did not explore other values of D. The encoders for both images and audio are convolutional networks which produce the parameterization (the mean and log variance vectors) of the posterior distribution over embeddings. The audio encoder uses a similar architecture as the encoder portion of Hsu et al. (2017a)'s variational autoencoder for 80 dimensional log Mel-filterbank speech. The specific architectures for the convolutional image and audio encoders is given in Table 3.1. A

[3] $\left(\frac{1}{2} - \frac{1}{2}\cos\left(\frac{2\pi n}{N}\right)\right)^{0.85}$, available in the Kaldi speech recognition toolkit (Povey et al. 2011).

[4] Training does not depend on explicit class labels except insofar as pairing audio and image inputs based on their ground truth digit labels.

Image Encoder	Audio Encoder
3×3 conv., 64 filters, same padding	$1 \times F$ conv., 64 filters, same padding.
ReLU	tanh
	BatchNorm (Ioffe and Szegedy 2015) 64 channels
3×3 conv., 2×2 strides, 128 filters, same padding	3×1 conv., 2×1 strides, 128 filters, same padding
ReLU	tanh
	BatchNorm (Ioffe and Szegedy 2015) 128 channels
3×3 conv., 2×2 strides, 256 filters, same padding	3×1 conv., 2×1 strides, 256 filters, same padding
ReLU	tanh
	BatchNorm (Ioffe and Szegedy 2015) 256 channels
Flatten to vector	Flatten to vector
512 unit fully connected	512 unit fully connected
ReLU	ReLU
	BatchNorm (Ioffe and Szegedy 2015) 512 channels
256 unit linear output	256 unit linear output

Table 3.1: Neural architectures used for the image and audio encoders for experiments with TIDIGTS/MNIST. The audio encoder architecture is based on the encoder used in Hsu et al. (2017a). Note that for the audio encoder, $F = 80$ mel-filters. For the 256 unit linear output, 128 units are for μ and 128 for $\log \sigma^2$.

weight decay (λ_{WD}) of 10^{-6} is used for all convolutional and fully connected layers. The initial learning rate was 10^{-5} which was decayed by a factor of 0.9 every 10 epochs. The Adam learning rate scheduler algorithm (Kingma and Ba 2014) was used with $\beta_1 = 0.95$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. 128 distinct image-audio pairs were used for each batch. After processing each image or audio input through the respective encoder to produce a posterior distribution, 16 embeddings were sampled per input.[5] This produced a total of 2,048 image-audio embedding pairs in each batch.

Negative sampling was performed by selecting one of the other 2,047 sample pairs in the batch. While it would at first seem reasonable to disallow negative samples for a training pair to be drawn from the same underlying digit class, such a mechanism implies a ground truth digit labeling of all examples within a batch. In other words, the knowledge of which negative example pairs *not* to sample is equivalent to the network possessing an oracle with knowledge of which audio/visual sample pairs within a batch were drawn from the same underlying digit class. This oracle would allow the network to trivially recover the ground truth digit labeling of all examples within a batch. In an effort to avoid this, I allow negative samples to be chosen from any digit class regardless of the initial example's digit class. Empirically, I found that

[5] Positive image-audio embedding pairings were established by matching corresponding sampled embeddings for each input.

the weight of the positive examples can easily overcome the “contradictory” signals introduced by this sampling scheme, allowing the model to produce a semantically rich embedding space.

The model was trained for 100 epochs. In contrast to the other models in this thesis, which were implemented using PyTorch (Paszke et al. 2017), this model was implemented using TensorFlow (Abadi et al. 2015). An epoch was defined as the number of batches required to cover all training examples in the larger of the two datasets (MNIST) exactly once. Training required about 35 minutes on an NVIDIA TitanX GPU.

3.6 Results

To analyze the learned semantic space, I sampled embeddings for inputs from the unseen test set, sampling 16 samples per input point. I ran K-means clustering with $k = 10$ and calculated the cluster purity of the resulting clusters, defined as:

$$\frac{1}{N}\sum_{i=1}^{k}\max_{j=1\ldots k}\left(|c_i \cap y_j|\right)$$

where c_i is the set of all points in cluster i and y_j is the set of all points of class j (their ground truth digit label). This metric represents the accuracy of a classifier which classifies a point, x, according to the majority class of the cluster whose mean is closest to x using euclidean distance.

I then used a subset of 2,152 sample embeddings (1,076 from images, 1,076 from audio) and performed a classification task to predict the original input point’s modality from the embeddings using a support vector machine (SVM) with a Gaussian RBF kernel. 1600 examples were used for the training set and the remaining were used for the test set. I used 3-fold validation to select a C value for the SVM. Comparing the modality classification test accuracy to the prior on modality ($\frac{1}{2}$) allows us to gauge the extent to which the embeddings are modality-invariant. Perfectly modality-invariant embeddings would result in a test accuracy of $\frac{1}{2}$ for the modality

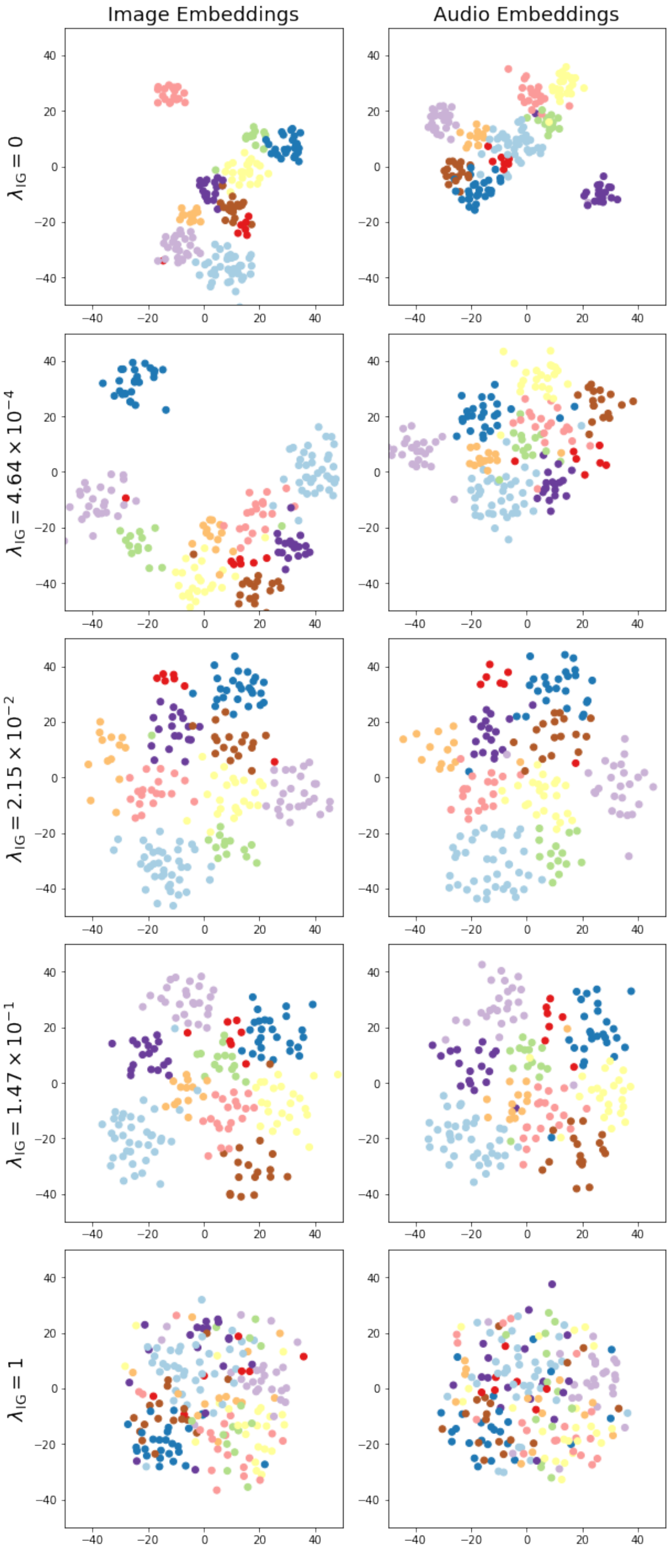


Figure 3-4: 2 dimensional t-SNE projections of 128 dimensional embeddings produced from using various weights, λ_{IG}, for the KL divergence regularization term.

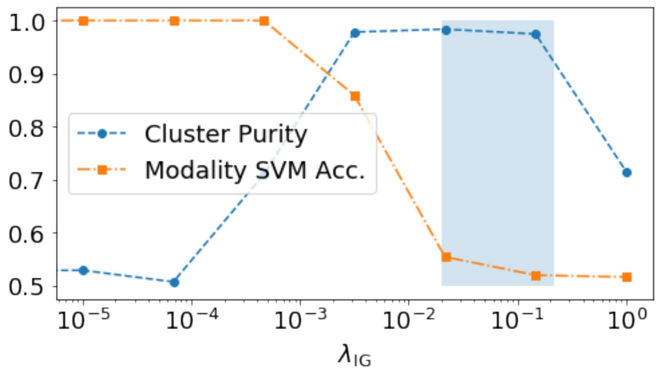


Figure 3-5: Effects of tuning the weight, λ_{IG}, of the KL divergence regularization term. The shaded region is considered ideal: modality classification is nearly random and cluster purity reaches its peak.

classification task.

I evaluated the effect of λ_{IG} on the cluster purity and modality invariance of the embeddings learned by the model. Results from using the modality classifier and cluster purity analysis are shown in Figure 3-5 and Table 3.2.

In addition, I used 200 samples per modality to compute a two dimensional t-SNE projection[6] of the embeddings produced by each hyperparameter setting. I plotted these samples in Figure 3-4 and colored them according to class label. For both cells in a row, the same t-SNE model was used, so the embeddings for both modalities were projected into the same two-dimensional space.

The additional $\mathcal{L}_{\text{IG}}$ term resulted in greater cluster purity, as shown in Figure 3-5. The lower cluster purity for $\mathcal{L}_{\text{Sim}}$ alone ($\lambda_{\text{IG}} = 0$) is visually evident in the first row of Figure 3-4: though there are clear semantic clusterings of samples from the same digit, there are typically two clusters per digit—one for images and one for audio. One possible explanation for why the cluster purity is low (0.525) for $\lambda_{\text{IG}} = 0$ is that when K-Means is performed with $k = 10$, k is about half the number of digit clusters in the embedding space (one for each digit-modality pair), resulting in K-Means clusters with members nearly evenly split between two digits. This finding shows that while using $\mathcal{L}_{\text{Sim}}$ alone, embeddings originating from the same modality may still be significantly

[6] t-SNE was selected over PCA for its ability to show relative pairwise distances (Maaten and Hinton 2008)

λ_{IG}	Cluster Purity	Modality SVM Acc.
0.00e+00	0.525	1.000
1.00e-05	0.542	1.000
6.81e-05	0.516	1.000
4.64e-04	0.707	1.000
3.16e-03	0.980	0.859
2.15e-02	**0.984**	0.554
1.47e-01	0.975	0.520
1.00e+00	0.679	**0.516**

Table 3.2: Cluster purities and modality classification accuracies for various values of λ_{IG}.

closer together than embeddings of different modalities, regardless of the similarity of semantic content. The 100% accuracy of the SVM in predicting the modality of embeddings when $\lambda_{\text{IG}} = 0$, as shown in Figure 3-5(a), further supports the finding that the embedding space produced from using $\mathcal{L}_{\text{Sim}}$ alone is not modality invariant.

In contrast, the embeddings produced when using $\mathcal{L}_{\text{IG}} = 2.15 \cdot 10^{-2}$ for training were only able to be classified by an SVM with 55.4% accuracy, as shown in Table 3.2. Although this metric is not the ideal 50% accuracy of truly modality-invariant embeddings, the embedding space produced using $\mathcal{L}_{\text{IG}}$ is much closer to being modality-invariant than the space produced by $\mathcal{L}_{\text{Sim}}$ alone.

Figure 3-5 shows that minimizing the divergence of the posterior over embeddings from the prior improves modality invariance. This could be due to the fact that the KL divergence represents the amount of information about an embedding conveyed by an input, and by limiting the amount of information, we force the encoders to filter out information. This is the same reason why variational autoencoders exhibit de-noising behavior (Kingma and Welling 2013). Since semantic information is important for minimizing $\mathcal{L}_{\text{Sim}}$, modality information tends to be filtered out before semantic information. For my model, Figure 3-5(ii) shows that $\lambda_{\text{IG}} \approx 2.15 \cdot 10^{-2}$ is the empirically observed ideal cutoff point at which increasing λ_{IG} to further limit the total information conveyed in the posterior begins to also to overly restrict the semantic information conveyed, resulting in a drop in cluster purity. Figure 3-4 shows this trend qualitatively. Row 1 shows sampled embeddings resulting from an underregularized model; row 3, well regularized; and row 5, over regularized.

3.7 Application to the Places Dataset

I ran preliminary experiments in which I attempted to apply this regularization technique to DAVEnet, trained on Places. Though the objective is stable, preliminary results suggest there is a significant trade-off between modality invariance and recall scores as λ_{IG} is tuned, suggesting that the two types of information are more complexly related for open-ended semantic concepts than for digits.

For stability in the training procedure, I found it necessary to make the following adjustments in the implementation. First, an epsilon term is needed for the log variance. Empirically, I found this necessary to avoid diverging to infinite losses. Second, a bias term is needed for the log variance. In neural network training, it is common to initialize the bias vector to zero. However, for the log variance component of the output, I initialized the bias to -8 (setting the initialize standard deviation of the predicted Gaussian to e^{-4}). For the more complex dataset, I found this change necessary for the margin similarity loss to decrease below random chance. Intuitively, this could be because the Gaussian noise with a log-variance of 0 (the default bias) overwhelms the signal-to-noise ratio for the similarity loss term at the start of training. By biasing the noise variance to start small, the model is effectively using the means as single points until it learns to increase the log-variance through stochastic gradient descent to decrease the $\mathcal{L}_{\text{IG}}$ term. By the end of training for $\lambda_{\text{IG}} = 10^{-4}$, log-variances were generally around -6.6 on average. Therefore, the log variance used in practice is:

$$\text{logvar}\left(x_v^{(i)}\right) = b\vec{1} + \log\left(\epsilon + e^{\log\sigma_v^2\left(x_v^{(i)}\right)}\right)$$

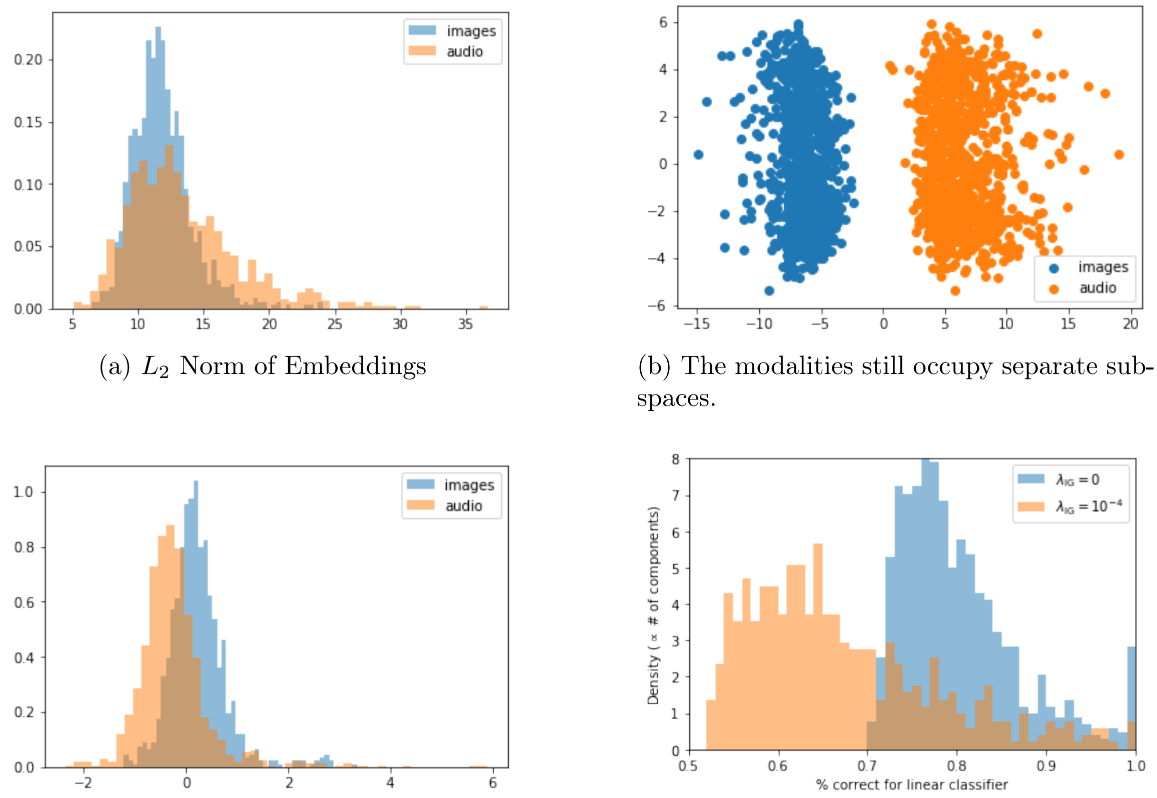
where b is the bias coefficient (-8) and $\epsilon = 10^{-8}$. The fact that both hyperparameter settings involve -8 is coincidental.

I ran experiments with four settings of λ_{IG}: 10^{-4}, 10^{-3}, 10^{-2}, and 10^{-1}, all with margin of 1, semi-hard-negative loss with coefficient 1, and learning rate of 10^{-3}. The image encoder was a ResNet-50 network pretrained on ImageNet. $\lambda_{\text{IG}} = 10^{-1}$ was not stable, and the loss quickly diverged. $\lambda_{\text{IG}} = 10^{-2}$ was stable, but failed to improve over random chance for modality retrieval due to over regularization. $\lambda_{\text{IG}} = 10^{-3}$ and $\lambda_{\text{IG}} = 10^{-4}$ both were able to increase above random chance for modality retrieval. $\lambda_{\text{IG}} = 10^{-3}$ obtained recall scores of 0.65 and 0.70 R@10 for image to caption and caption to image, respectively. $\lambda_{\text{IG}} = 10^{-4}$ obtained recall scores of 0.66 and 0.70 R@10 for image to caption and caption to image, respectively, which is slightly below the state of the art reported in Table 2.1 (0.69 and 0.74).

Despite recall decreasing slightly, modality invariance increases. Though the modalities still occupy separate subspaces (meaning that modality can still be perfectly classified), as shown in the PCA projection in Figure 3-6(b), the individual components of the embedding are more modality invariant than without regularization. Figure 3-6(a) shows the distributions of the L_2 norms of the embeddings are much more similar than without regularization (Figure 3-2(b)). In addition, the distributions for individual components (i.e. dimensions) of the embedding are generally aligned, as qualitatively shown for one component in Figure 3-6(c).

Quantitatively, I designed a procedure whereby I found the best linear classifier for the 1,000 validation points for each dimension. Since this classifier "stump" was classifying a finite number of points in 1 dimension, it was possible to try all $2N + 2$ classifiers and select the one with greatest accuracy. This best-classifier accuracy for each component is shown as a histogram in Figure 3-6(d). Note that the most modality-invariant component would have a classification accuracy of 0.5. Figure 3-6)d) shows that the $\lambda_{\text{IG}} = 10^{-4}$ model learns an embedding space with many more components with greater modality invariance than the $\lambda_{\text{IG}} = 0$ model.

Though there is some evidence suggesting that the regularization technique promotes modality invariance on the Places dataset, recall scores suffer, and the embedding space is not modality invariant to the same degree as for the models trained on MNIST/TIDIGITS. This suggests that modality information and semantic information are entangled to a greater degree for the open-ended semantic information present in the Places dataset as opposed to the discrete set of concepts present in MNIST/TIDIGITS.



(a) L_2 Norm of Embeddings

(b) The modalities still occupy separate subspaces.

(c) The distribution of activations for one component of the embedding.

(d) Component-wise modality classification stump accuracies. Height of bar represents number of components with the specific classification accuracy. 50% represent complete modality invariance.

Figure 3-6: $\mathcal{L}_{\text{IG}}$ regularization results of Imagenet-Pretrained Resnet-DAVEnet trained on Places with $\lambda_{\text{IG}} = 10^{-4}$.

3.8 Discussion: Modality Invariance for Open-Ended Semantic Concepts

One reason that the regularization technique proposed in this chapter might not extend well to Places is that open-ended semantic information, like scene descriptions, is not inherently modality invariant. There are degrees to which information can fit a concept, and there can be many concepts portrayed at once. For example, the caption, "A picture of a dog", might technically fit a closeup image of a dog lying on the grass, and it might also fit an image of a dog in the distance in a field, but in the latter there are other concepts conveyed in the image as well that might be more apparent. There may be other concepts represented in a modality that are important for a task like modality retrieval, but which prevent learning a strictly modality invariant representation without also losing important semantic information for modality retrieval. In other words, semantic concepts become blurred, and because of this it becomes less easy to separate semantic and modality information through regularization.

A possible method of counteracting the problem of entangled modality and semantic information might be to learn a factorized embedding space. In this setup, the embedding space is factorized into multiple subspaces. Some of these subspaces are regularized to be modality invariant, forcing any modality-specific information to be stored in a separate subspace. This would force the distribution over component-wise modality invariance (as shown in Figure 3-6(d)) to become a bimodal distribution with the components for the modality invariant subspace having high modality invariance (low classification accuracy) and the components for the non-modality invariance subspace having low modality invariance (high classification accuracy). This idea is similar to and inspired by the factorized variation autoencoder proposed by Hsu et al. (2017b).

3.9 Conclusion

In this chapter, my goal was to learn a joint modality-invariant semantic embedding space for speech and images in an unsupervised manner. I focused on spoken utterances and images of handwritten digits. I found that by sampling encodings rather than predicting them directly, and by regularizing the posterior distribution over embeddings, I was able to learn a more modality-invariant semantic embedding space. From an adversarial perspective, I was able to deceive an adversarial discriminator (the modality-classifying SVM) without the use of gradient reversal or any adversarial setup during training. This leads me to suspect $\mathcal{L}_{\text{IG}}$ may be a useful regularization term in other approaches to learning domain or modality invariant embeddings.

I then applied the technique to DAVEnet trained on the Places dataset. I found that though component-wise modality invariance improved, the embeddings were still not modality invariant in certain components. This finding suggests that for more complex datasets with open semantic concepts, modality and semantic information might be entangled to a greater degree than for datasets with closed and discrete concepts, like digits. Drawing inspiration from (Hsu et al. 2017b), I proposed a way to isolate non-modality invariant information in the embedding space using a factorized representation where modality invariance is encouraged for only part of the embedding.

So far in this thesis, the only modalities I have focused on have been images and spoken audio. In the next chapter, I explore a new dataset consisting of short three second videos. Using this new dataset, I explore the use of a new modality: video, which differs from images in that it has a time dimension and an ambient audio track.

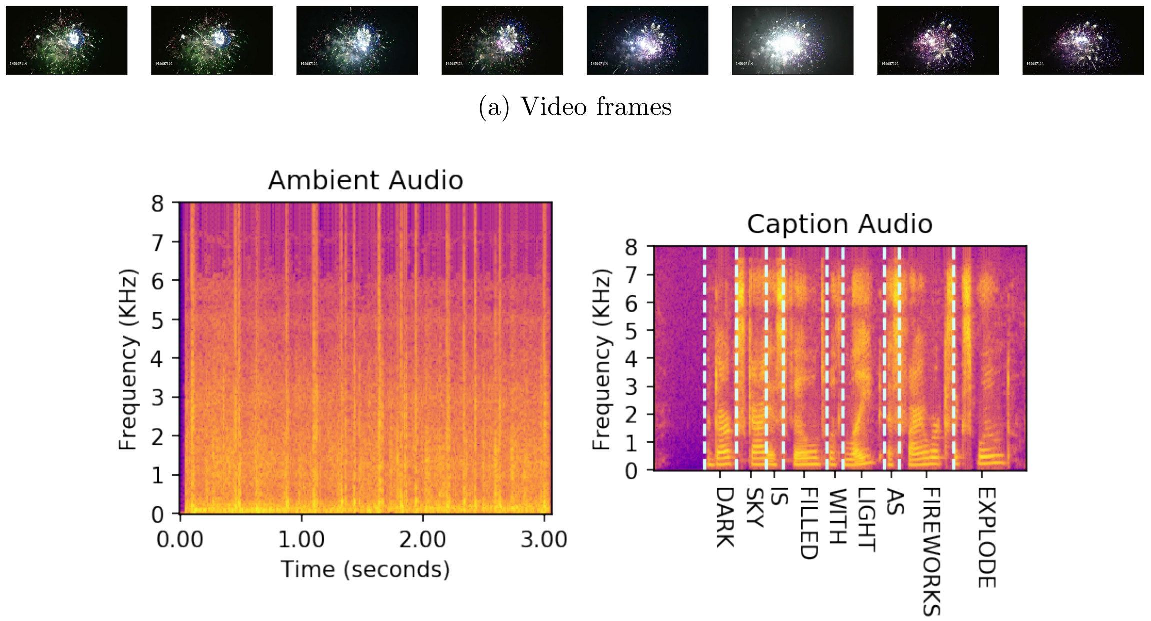
Chapter 4

Learning Actions from Captioned Videos

4.1 Introduction

Harwath et al. (2018a); Harwath and Glass (2017) have shown DAVEnet is able to learn hundreds of semantic concepts in an unsupervised fashion. However, these learned concepts typically correspond to objects, textures, materials, colors, backgrounds, etc. but not actions taking place in the images. There are a few possible explanations for this finding, for instance: (1) when someone describes an image, they are more likely to describe objects in the image and the setting of the image than actions taking place in the image, and (2) the Places dataset is better suited for object/scene recognition than action recognition. In this chapter, I experiment to see if either of these hypotheses could be the case. To do so, I turn to a new dataset: the Captioned Moments dataset (referred to simply as "Moments" for short), based on the Moments in Time dataset (Monfort et al. 2018).

The goal of this chapter is to use the Moments in Time video dataset, designed for a discriminative action classification task, in place of the Places dataset as a way to encourage DAVEnet to learn more actions and verbs. Eventually, if the network is capable of understanding a broader range of types of concepts, it may be possible to start learning relations between concepts, such as abstraction from known concepts



(a) Video frames

(b) Video ambient (left) and caption (right) audio

Figure 4-1: An example from the Captioned Moments corpus, with video from the Moments in Time corpus (Monfort et al. 2018)

to new concepts.

This chapter is organized as follows: first I introduce the Moments in Time dataset (Monfort et al. 2018) and our augmented dataset: the Captioned Moments dataset. This section includes examples from the dataset, information on how the dataset was preprocessed, and a link to the Github repository containing the metadata for the dataset and data loader scripts written for PyTorch (Paszke et al. 2017).

Second, I give a description of the DAVEnet models and architectures used for training and discuss opportunities for incorporating modalities not present in the image-based Places dataset. I then give results collected for various configurations of the model and training procedure. I conclude with a discussion regarding the opportunities and challenges of working with Captioned Moments dataset, including my recommendations for future work in the area.

4.2 Datasets

The Moments in Time dataset consists of three-second videos, each labeled with one of 339 actions which is taking place in the video (Monfort et al. 2018). There are 802,244 videos in the training set and 33,900 videos in the validation set. Moments in Time was designed for discriminative action classification and is typically evaluated using top-5 classification accuracy.

Using Amazon Mechanical Turk, we collected approximately 104,000 spoken captions for a subset of the videos in the Moments in Time dataset. Each caption describes one video from the Moments in Time dataset, and each video is usually only captioned once. 188 videos are captioned more than once[1] affecting a total of 379 utterances. For the purpose of partitioning and negative sampling, the fact that these videos were duplicates was ignored[2]. An example video/caption pair is shown in Figure 4-1.

Using the Google ASR API, we collected approximate text transcriptions of the spoken captions. The captions contain an average of 19.3 words, a median of 16 words, and a standard deviation of 9.5 words. These text transcriptions are used for evaluation purposes. In some experiments in this chapter, I train a modified DAVEnet model which uses text as an additional input modality. I explicitly specify whether text was used during training for a specific experiment.

I split the 104,000 utterances into two partitions: a development partition of 1,000 held-out utterances and a training partition of the remaining utterances. I performed this split as a random choice without replacement, selecting utterances uniformly at random (independent of speaker, label, or any other information). Table 4.1 shows the top 15 most common labels for each of the partitions.

Note that the Captioned Moments dataset is significantly smaller than the Places dataset, which has approximately 400,000 captioned images. There are also fewer speakers for the Captioned Moments dataset than the places dataset. There are

[1] at most three times

[2] two videos which were each captioned twice are split across the train/development split, but they have different captions in both cases.

Index	Label	# in Train	%
1	talking	1312	1.3%
2	standing	1219	1.2%
3	dancing	1147	1.1%
4	bicycling	1116	1.1%
5	sitting	1023	1.0%
6	laughing	955	0.9%
7	discussing	705	0.7%
8	running	704	0.7%
9	playing	679	0.7%
10	raining	675	0.7%
11	marching	628	0.6%
12	biting	614	0.6%
13	singing	592	0.6%
14	kneeling	591	0.6%
15	driving	560	0.5%

(a) Top 15 most common labels for 103,000 training utterances for Captioned Moments

Index	Label	# in Dev	%
1	standing	19	1.9%
2	dancing	16	1.6%
3	bicycling	13	1.3%
4	sitting	13	1.3%
5	talking	12	1.2%
6	squatting	11	1.1%
7	raining	10	1.0%
8	saluting	9	0.9%
9	crying	8	0.8%
10	burning	8	0.8%
11	walking	8	0.8%
12	hammering	7	0.7%
13	officiating	7	0.7%
14	discussing	7	0.7%
15	sprinkling	7	0.7%

(b) Top 15 most common labels for 1,000 development utterances for Captioned Moments

Table 4.1: Top 15 labels for both partitions of Captioned Moments

Speaker Index	# in Train	%
1	15991	15.5%
2	14760	14.3%
3	9324	9.0%
4	4466	4.3%
5	4289	4.2%
6	3772	3.7%
7	3369	3.3%
8	2869	2.8%
9	2781	2.7%
10	2655	2.6%

(a) Captioned Moments Train

Speaker Index	# in Dev	%
1	159	15.9%
2	132	13.2%
3	78	7.8%
4	47	4.7%
5	43	4.3%
6	34	3.4%
7	33	3.3%
8	31	3.1%
9	28	2.8%
10	26	2.6%

(b) Captioned Moments Dev

Speaker Index	# in Train	%
1	17620	4.4%
2	17399	4.3%
3	14435	3.6%
4	13573	3.4%
5	12887	3.2%
6	12834	3.2%
7	12710	3.2%
8	12031	3.0%
9	12026	3.0%
10	11830	2.9%

(c) Places Train

Speaker Index	# in Dev	%
1	50	5.0%
2	43	4.3%
3	38	3.8%
4	37	3.7%
5	36	3.6%
6	34	3.4%
7	31	3.1%
8	30	3.0%
9	30	3.0%
10	30	3.0%

(d) Places Dev

Table 4.2: Top 10 speakers with most utterances in each partition. For comparison, (c) and (d) are from Places.

2,683 speakers in the Places dataset (271 in the validation set) and only 780 speakers in the Captioned Moments dataset (154 in the validation set). Though this difference seems reasonable considering Captioned Moments is only one quarter the size of Places, there are two speakers disproportionately represented in the Captioned Moments dataset. Table 4.2 shows the speakers with the most utterances for each partition. Approximately 30% of the training set and development set for Captioned Moments consists of utterances from the top two speakers. In contrast, the top two speakers for Places only produced approximately 9% of the captions for the training and development sets. Another indicator that there is a disproportionate representation of the top two speakers in Captioned Moments is the raw count of utterances recorded for the top two speakers: in Captioned Moments, the counts are similar to the counts for the top speakers in Places, despite Captioned Moments only being a quarter of the size of Places. The disproportionate representation of the two speakers in Captioned Moments means the model might be more likely to overfit the over-represented speakers.

Only 71,475 of the videos for the Captioned Moments dataset have an ambient[3] audio track. Some models used in this chapter do not use the ambient audio, and for the ones that do, I explicitly specify how I handle the case when an audio track is missing.

Loading videos into a format suitable for training can be difficult. For one, there is the problem of data density. Since MPEG4 videos are stored as a compressed series of frame deltas, to convert the video into a series of dense image representations increases the memory footprint dramatically. In addition, to feed multiple frames through a deep image network dramatically increases the amount of GPU memory requires. For example, my three-frame models (which take three uniformly spaced frames as input) require four NVIDIA Titan-X Pascal GPUs with 12 GB of GPU memory each to train with a batch size of 128. For Places, only two Titan-X Pascal GPUs are required. Since (1) a sufficiently large batch size is necessary for the negative sampled margin rank loss objective, (2) intermediate states from the forward pass must be stored

[3] as opposed to caption

for the backward pass to avoid recomputing the forward pass, and (3) all pairwise modalities for the batch must be computed for the loss function, it is impossible to break up the batch into smaller chunks run serially in the forward pass to decrease the memory footprint without having to repeat forward computations during the backward pass.

Working with videos is also difficult because the most common way to decode the MPEG4 format is using FFmpeg (FFmpeg Developers 2017). To load video frames and audio into Python, one must either (a) launch a subprocess of the FFmpeg executable or (b) use the FFmpeg C library. Most available Python packages use (a), but I have found it too slow for online data loading during training, even when performed asynchronously. The few packages which have bindings for (b) (including OpenCV (Bradski 2000)) have often experienced segmentation faults, which could be due to low level race conditions when parallelizing the data loading pipeline using threads. For those reasons, I copy Monfort et al. (2018) and use FFmpeg to extract video frames as JPEGs and the ambient video audio as MP3s as a preprocessing step before training. Code to extract uniformly spaced frames is provided in the Captioned Moments Github repository.

4.3 Models

In the following experiments, I used a DAVEnet architecture with a ResNet-50 Image encoder, and residual audio encoder (see Appendix A). The models are trained with blended semi-hard negative loss and uniformly negative sampled margin ranking loss, using the objective function given in Equation 1.2.

I pool the outputs from the encoders to a single vector. Similarities are then computed between the pooled embeddings. Pooling to vectors enables the efficient use of semi-hard-negative sampling (Jansen et al. 2017), which requires the evaluation of all N^2 pairwise similarities in the mini-batch of size N. If similarities were computed between embedding maps rather than vectors, as performed in (Harwath et al. 2018a), semi-hard-negative sampling becomes computationally expensive. Currently, most experiments I ran on Captioned Moments were performed using vector models (pool, then compute similarity) with semi-hard-negative sampling due to the success of semi-hard-negative sampling. The experiments I report in this chapter all models are vector-based models.

I run a series of three experiments. First, I run basic vector DAVEnet models on Captioned Moments using a setup similar to Places using a random video frame at training time and the center frame at test time. The goal of this experiment is to provide a comparison to Places and set a baseline. Then, I explore how adding in video-specific information to the training process affects performance. In addition, I explore with varying effective video frame rate at evaluation time, when there are less GPU memory constraints.

Third, I describe preliminary experiments I ran using k modalities rather than just two. In these experiments, I treat ambient video audio as an entirely separate modality from the video frames. I also experiment with adding in the ASR textual caption as a fourth modality. In this setup, the sampled margin ranking loss[4] can be computed between modality pairs. For example, there is a sampled margin ranking loss between images and caption audio, caption audio and caption text, caption audio and video audio, etc. The sampled margin ranking losses are then added to compute the total loss. This formulation of the problem allows modality retrieval in multiple directions, for example: given a textual caption, find the most similar video audio, or given video audio, find the most similar video frames. These models are based on (Harwath et al. 2018b) in which the authors use spoken Hindi captions as an additional input modality and learn pairwise similarities between the six possible pairs of spoken English captions, Hindi captions, and images.

4.4 Experiments

In this section, I explore the use of videos in the traditional 2-way modality retrieval setting. For Places, the two directions were “image to caption” and “caption to image”.

[4] and optionally, the semi-hard-negative loss, although not used here.

				Caption to Image			Image to Caption		
# Frames	Dataset	Image Enc. Pre.	D.N. Pre.	R@1	R@5	R@10	R@1	R@5	R@10
0	Random Baseline			0.001	0.005	0.010	0.001	0.005	0.010
1	Places 100K			0.008	0.041	0.088	0.014	0.057	0.094
1	Places 100K	✓		0.075	0.225	0.339	0.084	0.231	0.338
1	Places			0.142	0.360	0.478	0.113	0.323	0.442
1	Places	✓		0.262	0.581	0.703	0.197	0.522	0.661
1	Moments			0.006	0.025	0.044	0.009	0.027	0.043
1	Moments	✓		0.025	0.095	0.148	0.037	0.110	0.177
1	Moments		✓	0.045	0.142	0.207	0.037	0.127	0.203
1	Moments	✓	✓	0.089	0.263	0.365	0.083	0.254	0.349

Table 4.3: One-frame, no audio vector experiment results, comparing Places and Captioned Moments. On the left, the columns are: number of frames, dataset, image encoder pretrained (on ImageNet), DAVEnet pretrained (on Places).

For Captioned Moments, they are “video to caption” and “caption to video”. However, for consistency across the datasets, I refer to the “video to caption” and “caption to video” directions as “image to caption” and “caption to image”, respectively.

4.4.1 Comparing Performance to Places

Since Places is significantly larger than Captioned Moments, I ran experiments with a subset of the Places dataset I call “Places 100K”, containing 100K training utterances and using the same validation set as used for Places. Table 4.3 shows the results of the experiments. Note that the recall performances are dataset-specific; that is, there is a different Places-specific validation set for Places models than Captioned Moments models. Without pretraining, the Captioned Moments model surpasses the random baseline, but barely, with an average R@10 score of 0.043. In contrast, the Places 100K model obtains an average R@10 score of 0.091 without pretraining (albeit on a different validation set). It seems that 100K utterances is not a sufficient amount of training data to obtain satisfactory performance for a non-pretrained model. In contrast, when trained on the full Places dataset, the non-pretrained model obtains an average R@10 score of 0.46.

For the models with the pretrained image encoder (pretrained on ImageNet), the results are the same: the Places 100K model performs approximately twice as well as the Captioned Moments model, and the the full Places model performs almost twice as well as the Places 100K model. It seems likely that the models with less data are

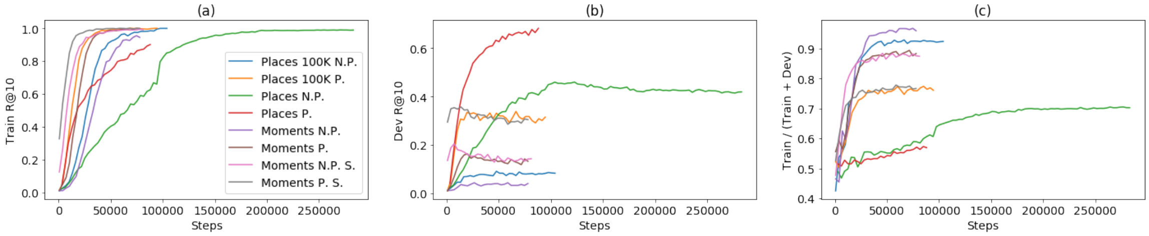


Figure 4-2: All plots show evaluation statistics over time during the training process, measured in number of minibatches of size 128. (a) shows the average R@10 recall on a 1000-utterance subset of the training data. (b) shows the average R@10 recall on the validation set. (c) shows the quantity training recall / (training recall + validation recall), used to assess overfitting. The closer the metric is to 1, the more the model is overfitting. If training and validation performance is the same, the metric is 0.5. P. stands for pretrained, N.P. for non-pretrained, and S. for seeded (pretrained DAVEnet on Places). Note that Captioned Moments and Places models are being evaluated on different datasets.

overfitting the training set. To check this hypothesis, I tracked the average R@10 score for the validation set and a 1000 utterance subset of the training set during training. The results are shown in Figure 4-2. Notice how all models approach 1.0 recall on the training set as training progresses. The smaller training sets approach 1.0 much more quickly and are most prone to overfitting. In addition, using the proportion metric shown in Figure 4-2(c), non-pretrained models overfit to a greater degree than pretrained models.

To see whether the poorer performance on Captioned Moments was an issue of training the model or simply a more difficult evaluation set, I evaluated each of the models on the both validation sets (Places and Captioned Moments). The results (only for R@10) are reported in Table 4.4. Notice that the model with an image encoder pretrained on ImageNet then trained on Places and fine-tuned on Captioned Moments performs the best on average across the two validation sets. The most interesting result is that the Places model with a pretrained image encoder outperforms the Captioned Moments model with a pretrained image encoder on the Captioned Moments validation set, suggesting that the overfitting due to the size of the training set is severely limiting the Captioned Moments model's modality retrieval ability.

Next I experiment with adding video-specific information to see if recall improves.

# Frames	Training Dataset	Image Encoder Pretrained	DAVEnet Pretrained Places	Evaluation Dataset			
				Places		Captioned Moments	
				I2A R@10	A2I R@10	I2A R@10	A2I R@10
1	Places 100K			0.088	0.094	0.025	0.033
1	Places 100K	✓		0.339	0.338	0.095	0.101
1	Places			0.478	0.442	0.133	0.138
1	Places	✓		**0.703**	**0.661**	0.270	0.237
1	Captioned Moments			0.024	0.028	0.044	0.043
1	Captioned Moments	✓		0.086	0.092	0.148	0.177
1	Captioned Moments		✓	0.262	0.216	0.207	0.203
1	Captioned Moments	✓	✓	0.421	0.397	**0.365**	**0.349**

Table 4.4: Results of evaluating one-frame no audio models on both validation sets. Only R@10 scores are reported.

4.4.2 Training with Video-Specific Modalities and Dimensions

I ran a series of three-frame video experiments. In these experiments, I used three evenly-spaced frames from the video as input. I used the image encoder to encode each frame independently. In this section, after encoding the video frames independently using the image encoder, I pooled the frame maps along the time axis, pooled them spatially, and finally computed the similarity score with the pooled audio caption embedding using the dot product. Three frames, though seemingly small, was the most I could use while fitting the batch into GPU memory using four NVIDIA Titan-X Pascal GPUs. Note that all models in this section were seeded with the weights from a DAVEnet trained on Places with an image encoder which was first pretrained on ImageNet.

In addition, I experimented with adding in ambient video audio as additional input. For these models, I used a shallower version of the residual audio encoder used for captions: it only uses 4 residual blocks. Note that the weights in the video audio encoder are not shared with weights in the caption audio encoder. After the video audio is encoded and average pooled, it is concatenated to the pooled video vector to make the video embedding. The ambient video audio embedding vector is 128-dimensions while the video encoder output is 1,024; therefore, the video embedding is a 1,024 + 128 = 1,152 dimensional vector. Correspondingly, the output of the caption audio encoder is 1,152 dimensions. The similarity score is computed by taking the dot product of the combined video embedding and the pooled caption audio embedding.

Table 4.5 shows that the addition of modality-specific information during train-

	Video		Caption to Image			Image to Caption		
# Frames	Audio	Frame Pooling	R@1	R@5	R@10	R@1	R@5	R@10
1		-	0.089	0.263	0.365	0.083	0.254	0.349
3		Max	0.080	0.261	0.372	0.095	0.280	0.400
3		Mean	**0.122**	0.293	0.396	**0.103**	0.288	0.377
3	✓	Max	0.078	0.268	0.382	0.085	0.281	**0.416**
3	✓	Mean	0.112	**0.300**	**0.419**	0.101	**0.289**	0.414

Table 4.5: Modality retrieval results from adding modality-specific information. The pooling procedure was: pool frames temporally, average pool frames spatially, then compute similarity. The temporal frame pooling method used at both training and evaluation time is specified in the "Frame Pooling" column.

ing, particularly additional frames, improves recall performance. Adding video audio tends to improve R@10 performance marginally, but this improvement could be noise due to stochasticity during the training process. To evaluate the significance of the improvement, I lesioned the video audio components of the embeddings for the model trained with video audio at evaluation time. Figure 4-3 shows that the lesioned model performs nearly as well as the non-lesioned model in all cases, suggesting that the video audio component of the embedding does not benefit recall. However, the audio could help provide additional grounding during training, resulting in the improved recall performance when audio was added in Table 4.5. Since R@10 performance improved about 1% for both pooling procedures when audio was added during training, it leads me to think the improvement is not coincidental; rather, the addition of the video modality helps to provide additional grounding during training.

Figure 4-3(a) shows that increasing the number of frames used at evaluation does not necessarily improve performance. In fact, when max temporal pooling is used, it decreases performance. This could be because the model becomes more susceptible to noisy outliers in activations in one of the frames as more frames are added. The probability there is an outlier increases as the number of frames increases. Average pooling does not suffer from this problem and, in fact, benefits the additional frames, improving approximately 2% from the 1 to 15 frame case as showing in Figure 4-3(b).

Note that since video frames are encoded independently and then pooled temporarily using max or average pooling, the current model has no notion of a change

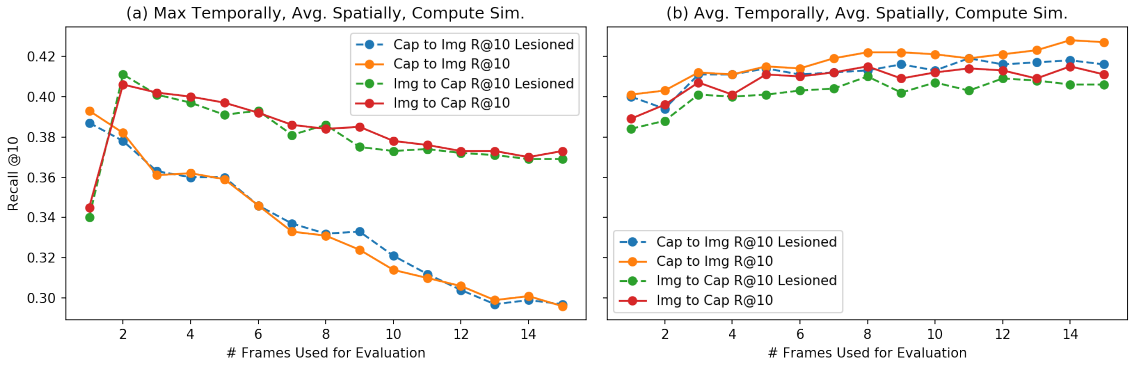


Figure 4-3: The effect on recall when increasing the number of frames used at evaluation time. Two different models were used for (a) and (b) with the only distinction being the temporal frame pooling procedure used (both at training time and evaluation time). In both cases, three frames were used during training.

in the content of video over time. In other words, the increase in performance with additional frames is almost solely due to (a) additional training data and (b) better visibility for objects occluded or poorly oriented in one frame. To allow the model to learn to account for change over time, a volumetric convolution along the spatial and time axes could be used; however, I did not run experiments to this end.

The experiments in this section support the claim that additional video-specific information, such as the temporal dimension for visual information and the ambient video audio, contain information which can improve DAVEnet's performance on modality retrieval tasks.

4.5 Experiments Across k Modalities

I briefly experimented experimented with k-way modality models before switching my focus 2-modality models. The models in this section all used a VGG16 (Simonyan and Zisserman 2014) pretrained on ImageNet as the image encoder and the traditional DAVEnet map convolutional audio encoder, described in Appendix A. All encoders used tanh as the final output function, constraining activations to the range -1 to 1. The reason for using tanh was to avoid the asymmetry of using linear output for one encoder (the image encoder) and ReLU output for another encoder (the audio en-

	A2I	I2A	A2T	T2A	A2V	V2A	I2T	T2I	I2V	V2I	T2V	V2T
IA	0.287	0.259	-	-	-	-	-	-	-	-	-	-
IT	-	-	-	-	-	-	**0.371**	0.408	-	-	-	-
IAV	0.263	0.246	-	-	0.028	0.050	-	-	0.111	0.147	-	-
ITV	-	-	-	-	-	-	0.348	0.392	0.107	0.155	0.064	**0.082**
ITAV	**0.289**	**0.266**	**0.825**	**0.796**	**0.051**	**0.070**	0.365	**0.412**	**0.112**	**0.157**	**0.067**	0.073

Table 4.6: Recall @10 from k-way modality experiments. “I” corresponds to video frames, “V” to ambient video audio, “A” to caption audio, and “T” to caption ASR text.

coder), as empirically are the best performing output activations when working with only images and caption audio. The margin is set to 0.1, which I found works better for the tanh output activation than the traditional margin of 1. In addition, the models are trained with sampled margin ranking loss alone, rather than blended sampled margin ranking loss and semi-hard-negative loss. This was due to the fact that sampled margin ranking loss and the ResNet-based architecture were not available at the time the experiments were run. Only one random frame from the video was used during training time, and the center frame was used at evaluation time. Caption audio and video frame encoders were initialized with weights from a DAVEnet pre-trained on Places. For models which used caption ASR text, the encoder consisted of an embedding layer mapping words to 200-dimensional embeddings, followed by two 1D convolutions with kernel size 3 and stride 1.

Table 4.6 shows the results of the experiments. The table shows the resultant change of recall as video audio and caption ASR text are added as additional modalities. The only modality which consistently improved recall was the caption text. In fact, the addition of video audio tends to decease recall. This could be due to overfitting in one modality pair.

In practice, I found the k-way modality pair setup poses many challenges. Namely, it can be difficult to balance the gradients from the multiple modality pairs. Oftentimes, the easier direction (audio to text, text to audio, for example) overwhelm the gradient for the loss, causing the embedding space to be well suited for that one direction but no other directions. One possible way to overcome this might be to use a factorized embedding space. In this setup, certain components of the embedding for one modality would be designated for computing similarity with another modality. In other words, there would be a separate embedding space and similarity function for every pair of modalities. Transfer learning could still occur, as the encoder network leading up to the final split would be shared across the factorized subspaces.

My preliminary experiments using k-modalities did not obtain satisfactory results compared to the 2-modality models described in the previous sections. However, with more time and the use of certain modeling techniques, like a factorized embeddings space, future work could certainly obtain better results for k-way modality modeling. I conclude with a summary of the pros and cons of k-modality modeling versus 2-modality modeling. First the pros:

1. For modality retrieval, it is possible to query any modality given any other modality. For example, I can query for the most similar video frames to an ambient audio recording.

2. In theory, the objective seems better suited for transfer learning and grounding, as described in (Harwath et al. 2018b).

The cons:

1. It can be difficult to tune the learning rate for each direction of retrieval to avoid easier modality-pairs dominating the gradient.

2. It is not clear how to query for a modality given multiple modalities. For example: “what is the most similar audio caption to the given video image *and* ambient audio?”

4.6 Word and Concept Learning Experiments

In this section, I describe the technique I used to assess the concepts learned by a model for a particular dataset. The technique is called neural dissection, proposed by Bau et al. (2017); Zhou et al. (2017). The process is also quite similar to the process used in Section 5.2 of (Harwath et al. 2018a), except that I do not use the image embeddings or similarity scores; only the audio embeddings. This has the added

Only Places (forgot 17 words)	Both (retrained 99 words)	Only Moments (learned 17 words)
around, castle, dining, garden, lights, market, office, painting, photo, restaurant, side, sign, stairs, station, tall, tower, video	area, baby, background, baseball, beach, black, blue, boat, boxing, boy, brick, bridge, brown, building, buildings, bushes, car, cars, chairs, child, children, church, city, clouds, dancing, dark, desk, dirt, distance, door, dressed, fence, field, fire, floor, flowers, fountain, girl, glass, grass, green, ground, house, inside, jacket, kitchen, large, machine, man, men, mountain, mountains, night, ocean, orange, outside, parking, people, person, plants, playing, pool, red, river, road, rock, rocks, rocky, room, shirt, sitting, skies, sky, small, snow, stage, standing, stone, store, street, table, tables, top, track, tracks, train, tree, trees, walking, wall, walls, water, white, window, windows, woman, wooden, yellow, young	ball, bicycle, camera, dog, game, girls, group, hands, he, picture, running, says, talking, two, watching, wearing, wood

Table 4.7: Words for which a detector was learned, based on network dissection (Bau et al. 2017; Zhou et al. 2017) analysis with transcript forced-alignments. The model used for Captioned Moments was fine-tuned on Places. Both models were evaluated on the concatenated validation datasets from Places and Captioned Moments, so as to have the same evaluation vocabulary for network dissection.

benefit of not requiring pixel-level annotations for the images, which is required for the technique described in Section 5.2 of (Harwath et al. 2018a) but which is not available for the datasets of interest in this chapter. In addition, pixel-level annotations are inherently biased towards objects, and one goal of this section is to study whether training DAVEnet on the Captioned Moments dataset will cause the recognizer to learn more action words.

In contrast to (Harwath et al. 2018a), the technique used here does not seek to prove that the image encoder and audio encoder are agreeing upon a concept. Rather, it merely shows which words activate specific components of the audio embedding. Here, “activate” means that the component’s activation was within the top 0.5% of activations for the unpooled audio embedding in that dimension. Image embeddings which also respond strongly in that dimension may be recognizing a visual stimulus corresponding to that word, but they more generally will respond to a stimulus correlated to the presence of the word in the caption. For example, if the word the audio encoder responds to is “street”, the image encoder might respond to a street lamp or traffic light, despite them not explicitly being streets. Likewise, for action verbs like “running”, the image architecture might respond to specific visual stimuli, like a racing bib, or the runner’s legs.

Using the audio embedding map output from DAVEnet before pooling to a single vector, I generate a set of embedding vectors paired with words based on the force alignment of the Places Google ASR transcripts of the audio captions. Note that these embedding-word pairs are downsampled by a factor of eight from the input Mel-filterbank, having an effective windows size of 200 milliseconds and a frame shift of 80 milliseconds. The receptive field size due to convolution is approximately 1.3 seconds. Because of the downsampling, it is common for there to be multiple words associated with a single time step in the embedding map. For that reason, I calculate IOU scores on a per-word basis. That is, I take the bit vector corresponding to the presence of a single word at each time step in the entire audio corpus (concatenated along the time axis) and calculate the intersect-over-union with the bit vector corresponding to the activation of a particular component of the embedding. Let:

$$W(w,t) := \text{word } w \text{ is present at timestep } t$$

IOU can then be defined:

$$\text{IOU}_{w,d} = \frac{\sum_t^T [[W(w,t) \wedge A_{t,d} > \text{percentile}_{t'}(A_{t',d}, \tau)]]}{\sum_t^T [[W(w,t) \vee A_{t,d} > \text{percentile}_{t'}(A_{t',d}, \tau)]]}$$

where T is the number of timesteps in the corpus, d is the dimension of interest in the embedding, and w is the word of interest. τ is the threshold for activation. I set $\tau = 99.5$th percentile for my experiments, in accordance with Zhou et al. (2017). IOU can be though of as the ratio of co-occurrences to all occurrences, taking values between 0 and 1. It can also be thought of as the fraction of time when one of the events[5] occurs, both of them occur.

Using the IOU score, it is possible to see if the activation of individual components is correlated to the presence of certain words in the transcript. To do so for a particular component of the embedding, I select the words with the top IOU scores for the component. As an additional threshold, I only consider a word-component pair a detector if the IOU score exceeds 0.05 (5% of all occurrences of the word or the component are co-occurrences).

Note that IOU's should not be thought of like an accuracy. Due to the thresholding of activations, only the top 0.5th percentile of activations for a component are considered "activated" for the purpose of IOU calculation, but the presence of a word could still result in a high activation for that component. In addition, if the firings occur at the middle frame of the word, but not in the surrounding frames, IOU may still be low. For those reasons, the IOU scores should not be considered accuracy metrics but rather as a metric for comparing the relative strength of detectors.

I ran the IOU analysis on two models: the first was trained on Places and the second was initialized with the parameters of the first and then fine-tuned on Captioned Moments using three frames and audio, as described in Section 4.4.2. Note that the

[5] the activation of an embedding component or the presence of a word

Word	IOU Before	IOU After	Rank Before	Rank After	# Passed	# Passed By
talking	0.037 (1)	0.104 (**1128**)	145 / 160	48 / 160	97	0
hands	0.028 (1022)	0.080 (937)	157 / 160	73 / 160	84	0
baby	0.073 (22)	0.163 (876)	84 / 160	25 / 160	59	0
wood	0.041 (1023)	0.069 (1023)	139 / 160	85 / 160	56	2
bicycle	0.036 (866)	0.057 (866)	149 / 160	98 / 160	52	1
ball	0.032 (277)	0.054 (49)	152 / 160	104 / 160	49	1
dog	0.047 (276)	0.077 (**1119**)	122 / 160	76 / 160	48	2
camera	0.048 (1004)	0.077 (822)	121 / 160	77 / 160	47	3
game	0.037 (749)	0.055 (97)	146 / 160	103 / 160	45	2
large	0.078 (691)	0.121 (691)	78 / 160	38 / 160	41	1
he	0.030 (188)	0.052 (188)	154 / 160	115 / 160	40	1
running	0.036 (301)	0.053 (64)	148 / 160	110 / 160	41	3
she	0.028 (1002)	0.049 (282)	156 / 160	119 / 160	38	1
wearing	0.042 (187)	0.056 (16)	136 / 160	99 / 160	41	4
two	0.037 (231)	0.053 (542)	147 / 160	111 / 160	38	4

Table 4.8: The 15 words which moved forward the most in the sorted list of max IOU-ranking when fine tuned on Captioned Moments. Green words correspond to "learned" words and blue words to "remembered" words. Only words with IOU greater than 0.04 were considered. Words with IOU greater than 0.05 were considered known. The parenthesized integer next to the IOU score corresponds to the dimension of the embedding with that particular IOU score. Dimensions greater than 1024 are shown in red and correspond to ambient video audio dimensions. Notice that "talking" and "dog" have detectors in dimensions 1128 and 1119 (ambient video audio dimensions) of the embedding. The full table is given in Appendix B.

IOU scores were calculated from the same validation set: the concatenation of the Places validation set and the Captioned Moments validation set.

There were 456 total detectors and 319 dimensions containing detectors for the Places model. There were 496 total detectors and 361 dimensions containing detectors for the Captioned Moments model. The Places and Captioned Moments models both had detectors for 116 unique words, but each model had detectors for different words. See Table 4.7 for a list of the words learned/forgotten when the model is fine-tuned on Captioned Moments. In addition, Appendix B gives a more detailed list of word detector changes between the pretrained and fine-tuned models.

Notice that certain "scene" words like "restaurant", "station", "market", "office", "castle", "garden", and "dining (room)" are forgotten when the model is trained on the Captioned Moments dataset. However, action words, like "running", "talking", "watching", and "wearing" are learned. More precisely, there were component-wise detectors exceeding the IOU threshold for the "forgotten" words before fine-tuning that weren't present after fine-tuning, and there were "learned" words with component-wise detec-

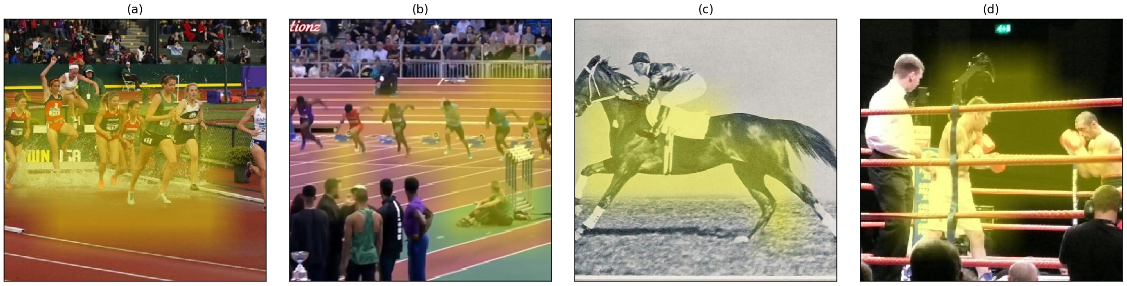


Figure 4-4: Locations where component-wise detector for dimension 64 is activated (activation of embedding is greater than 99.5th percentile). Dimension 64 is a detector for the word "running" for the audio encoder, with an IOU score of 0.053. For visualization, the boolean mask representing detector activation was upsampled to the size of the image using bilinear interpolation, resulting in non-binary values of the overlay's opacity channel.

tors after fine-tuning that did not have detectors before fine-tuning.

I also performed a ranking-based analysis of the learned words. Since IOU cutoffs for known words are somewhat arbitrary, instead I sorted the list of words by their max IOU scores over all dimensions. I sorted the words based on their IOU scores before and after fine tuning and tracked how the position of the word in the list changed after fine tuning. Table 4.8 shows the top 15 words which moved up the most in the sorted list, passing many words and being passed by few. As you can see, action words like "talking", "running", and "wearing" are in the top 15. Also interesting: the detectors for "talking" and "dog" are in ambient audio dimensions of the video embedding. This suggests that the model has learned a rudimentary form of voice activity detection in an unsupervised fashion, and might be able to recognize a dog bark or whine. The full version of Table 4.8 is located in Appendix B.

Since the network dissection so far has been performed on audio embeddings alone with the goal of identifying recognized words, I look to the images/videos to qualitatively gauge if the recognized words correspond to relevant regions of the images/videos. In lieu of a pixel-wise labeling for the datasets of interest, the quantitative analysis to this end performed in (Harwath et al. 2018a) is not relevant here. Instead, I used the technique of thresholding a component of the embedding's activation on the image embeddings. Just like when I thresholded activation for audio embeddings, I set the threshold for activation to be the 99.5th percentile. From the audio IOU analysis, dimension 64 (out of 1024) had a detector for the word "running" with an IOU score 0.053. I selected the images with a large number of spatial regions with dimension 64 activated, based on the image embedding thresholding. The results are shown in Figure 4-4, with the yellow overlay corresponding to the regions in which dimension 64 of the embedding is activated above the 99.5th percentile threshold.

Analysis was also performed using precision/recall/F_1/$F_{0.5}$ analysis using the activation of a component as a prediction. Using $F_{0.5}$ score, results were quite similar to using IOU score, and the use of IOU score is consistent with Bau et al. (2017); Zhou et al. (2017), so IOU results were used. In general, precision was high and recall was low, likely because there were short spikes of activation in the middle of words, and the threshold for activation is quite high (the 99.5th percentile).

4.7 Chapter Summary

The audio IOU analysis gives quantitative support that the network recognizes the word "running" while the images give qualitative support that the network understands the concept "running", at least to some degree. Hypothesis (2) from Section 4.1 is supported by the fact that the network learns detectors for new action words when fine-tuned on the Captioned Moments dataset. That is, the fact that DAVEnet learned few action words when trained on Places was in part due to the dataset. However, there are still a disproportionately large number of object words recognized by the network, suggesting that hypothesis (1), that captions are more likely to describe objects present in an image, also has merit.

The Moments in Time dataset is quite large, having approximately 800K videos. Once Captioned Moments is expanded to have captions for a greater proportion of these 800K videos, the problem of overfitting the training set and requiring pretraining on another dataset should be resolved. Future work might focus on adjusting the learning objective to allow the model to be trained with a smaller batch size. This would help combat the constraint of GPU-memory, which is an impediment for fast model iteration since four GPUs are currently required to train the 3-frame models.

Another interesting direction to explore is to better explore DAVEnet's capacity to model retrieval for the directions "video frames to ambient audio" and "ambient audio to video frames" (without captions). This is similar to Aytar et al. (2016)'s work, except that instead of having categorical distributions over explicit objects/scenes which are regularized to be similar, a learned similarity function is used. Preliminary experiments I conducted exploring this task show that DAVEnet can attain at least 0.23 R@10 for both directions, but further experiments are needed.

Chapter 5

Conclusion

5.1 Summary of Contributions and Findings

In this thesis, I:

1. Identified and presented a solution to the problem of bias towards certain audio captions during modality retrieval,
2. Defined the property of modality invariance and proposed a regularization term to use during training to encourage the property in the learned embedding space,
3. Applied DAVEnet to a new dataset with additional modalities: the Captioned Moments dataset consisting of three second videos (Monfort et al. 2018), showing that training on this new dataset enables the learning of action-related concepts and concepts grounded to ambient sound.

In Chapter 2, I analyzed the behavior of DAVEnet on the modality retrieval task, finding that the network exhibited a bias towards certain audio captions during the image to caption retrieval task. One of these "favorite" audio captions were selected in the top 10 audio captions for over one hundred images. Meanwhile, over 500 audio captions were never selected as the most similar audio caption to an image. I proposed a technique to compensate for this bias at evaluation time: using the posterior inverse prior (Equation 2.1) in place of similarities.

A posterior probability distribution over retrieved captions given a queried image was estimated from the similarities using the softmax function. Then, the prior probability a caption is selected was estimated using an expectation over images in the training set. During modality retrieval, the similarity is replaced by the odds ratio of the posterior probability a caption is selected given an image divided by the prior probability the caption is selected. Performance for image to caption retrieval showed a relative improvement of about 20% R@1. In addition, the empirical distribution over selected audio captions was less skewed, suggesting the posterior-inverse-prior helped eliminate the problem of bias towards certain audio captions during modality retrieval.

In Chapter 3, I introduced the property of modality invariance for a semantic embedding space. I used a regularization term corresponding to the amount of information contained in an input to filter out information considered noise for the sampled margin rank loss objective. In the case of the TIDIGITS/MNIST dataset I used for my experiments, modality information was considered noise for the objective and filtered out, while semantic information (e.g. digit identity) was preserved. I ran preliminary experiments applying the regularization technique to DAVEnet trained on Places, and I found that the component-wise modality invariance increased without significantly impacting modality retrieval performance. I also discuss why it might be difficult to disentangle semantic and modality information for open-ended semantic concepts: certain concepts have inherently modality-specific properties, unlike digits. One way to account for modality-specific semantic properties might be to use a factorized embedding space, wherein some dimensions of the embedding are constrained to be modality invariant while others are allowed to contain modality-specific information.

In Chapter 4, I described the new Captioned Moments dataset: a spoken caption-augmented subset of the Moments in Time dataset (Monfort et al. 2018). I analyze properties of the dataset and note practical considerations for using it to train models. I trained a video frame-level DAVEnet on the dataset to compare the performance of models trained on Places with models trained on Captioned Moments. I found that models trained on Captioned Moments tended to overfit the training set fairly quickly, but to a similar degree to models trained on a similar-sized subset of Places. This finding suggests that more captions must be collected for the approximately 700K non-captioned videos available in Moments in Time before training non-pretrained DAVEnets on the dataset can perform satisfactorily.

I then experimented using multiple input frames from the video as well as the ambient audio track from the video. I found that performance improved as frames were added, using certain pooling methods. I also performed a word-learning experiments using network dissection (Bau et al. 2017; Zhou et al. 2017) that showed that when a model which was pretrained on Places was fine-tuned on Captioned Moments, component-wise detectors for action words like "running", "talking", "wearing", and "watching" were learned. In addition, I found that component-wise detectors for "talking", "dog", and "machine" were learned in ambient audio-specific components of the embedding, showing that the caption audio is being grounded to the ambient audio track for certain concepts. This is also a promising finding, as it suggests that the model might be learning a rudimentary voice activity detector in an unsupervised fashion.

5.2 Future Directions

For my work in Chapter 2, future research should explore how one might decrease bias toward captions during training rather than as a post-processing step. For example, one option might be to regularize the conditional entropy of the posterior distribution over retrieved captions given a queried image using the KL divergence from a uniform prior. This could have the same benefits as posterior-inverse-prior post-processing without breaking the maximum a posterior selection criterion.

For my work concerning modality invariance, future research should continue to explore the property of modality invariance and how to attain it in embedding spaces for open-ended semantic concepts, like objects and scenes. It may be the case that the regularization technique proposed is not well suited for situations where there are modality-specific components to some concepts that cause modality and semantic information to be entangled. In such cases, it may be necessary to learn a factorized embedding space or use an adversarial setup, like Saito et al. (2016), to achieve modality invariance. In addition, future work could explore the property's usefulness in a generative setting, such as for modality translation using the embedding space as a latent representation, similar to Hsu et al. (2017a;b)'s work using variational autoencoders to model speech.

For future work on Captioned Moments, the first priority should be to collect at least 300K more captions for the non-captioned videos in the Moments in Time dataset. This would help reduce the problem of overfitting and allow for a fair comparison with the full Places dataset using non-pretrained models.

After more captions are collected, new sampling procedures should be explored that allow a smaller batch size to be used. For example, if semi-hard-negative loss is not used, sampled margin rank loss can be reformulated using a small batch size by having batches of globally negative sampled triplets rather than sampling the triplets from the current, sufficiently large batch. Such a sampling procedure would remove the "sufficiently large" constraint on the batch size which makes it difficult to fit into memory the intermediate states for DAVEnet models which accept multi-frame video as input.

Additional word and concept learning experiments should be conducted on the Captioned Moments models, and on DAVEnet models in general. The word learning experiments I explored involved the analysis of component-specific detectors, but the model may understand complex concepts which aren't well represented by a single component in the embedding. For that, hierarchical clustering techniques, as used by (Harwath et al. 2018a), may be used. One interesting direction would be to cluster the embedding space in such a way that the clusters would maximize intercluster similarity, using the learned similarity metric. This would differ from the clustering analysis used in (Harwath et al. 2018a), since the similarity metric itself would be used as the criterion for "goodness" of cluster membership, rather than Euclidean distance in the high dimensional embedding space of the concatenated image and audio embeddings. Since similarity cannot be computed between two images or two audio components, a notion of "transitive intra-modality similarity" could be introduced; for instance, if computing the similarity between two images, one could find a path between the images via an audio caption which maximizes the sum of the similarities. For example:

$$S_{\text{intramodality}}(I_1, I_2) = \max_A \left(S_{I_1,A} + S_{I_2,A}\right)$$

where $S_{I,A}$ is the traditional inter-modality similarity between image I and audio caption A. There are likely ways to formulate this transitive similarity metric instead as an expectation over queried audio, using the probabilistic formulation of the similarity introduce in Chapter 2. This intra-modality similarity metric would be similar to the idea of transitive grounding introduced in Chapter 1 and depicted in Figure 1-1.

The use of topic modeling to model the audio embedding space should also be explored. This approach would allow the temporal context of audio embeddings to be better modeled. For example, in the Latent Dirichlet Allocation (LDA) approach (Blei et al. 2003), a spoken caption utterance would represent a document, and the document would be characterized as a mixture of topics. A likelihood distribution would be used to model the probability of an audio embedding frame given the topic. Using generative approaches such as LDA on the high level embedding space learned by DAVEnet might enable the learning of semantic relations between learned words.

5.3 Discussion

Multimodal concept learning is an exciting new area of unsupervised research. It has the potential not only to learn to recognize new words, but also to recognize visual and auditory stimuli that correspond to those words. This might show that the model possesses a much deeper understanding of the word and visual concepts, an understanding grounded in past experience of co-occurring stimuli. More over, due to the transitive nature of the pairwise similarity metric, the model has a notion of word-to-word similarity and image-to-image similarity through grounding with another modality. Unlike models like skip-grams or continuous bag of words, which look for intra-modality context to learn similarities in how words are used among other words, multimodal models use cross-modality context to learn similarities in how words are expressed. However, this is not to say that intramodality context is not important.

Imagine word-embeddings which capture not only how a word is used in a sentence and what words might be synonyms based on similar usage, but also what the concept the word refers to looks like, what it sounds like, and how it moves and interacts with its environment. Such a word-embedding could be constructed using the concatenation of DAVEnet embeddings correlated to words with the embeddings from a skip-gram model. Recent work by Chung and Glass (2018) has even shown that skip-gram models can be applied directly to speech, so it is not unreasonable to consider a speech embedding space which captures both cross-modality expressive information from DAVEnet embeddings as well as grammatical and intramodality information from a skip-gram model. Of course, a key component of these embeddings is that they not only contain information concerning object/scenes, but also actions (as I explored in Chapter 4) and object relations. Promoting modality invariance (as I explored in Chapter 3) might help denoise the embedding space of modality specific information which distracts from the more important semantic content. Finally, there may be a need to eliminate bias towards easier-to-recognize, commonly occurring concepts (as I explored in Chapter 2). Once constructed, this semantic embedding space may enable the creation of speech understanding models which can rival the performance of text-based natural language understanding, an outcome which is promising for extending natural language understanding to non-written oral languages.

The field of speech understanding is on the brink of learning its own written language. For many years, it has been the assumption of many researchers and engineers that in order to do any high-level natural language understanding on spoken language, one first had to convert speech to text and then train models to understand the text. DAVEnet is proving that wrong. DAVEnet has shown that its possible to uncover discrete concepts which align very naturally with discrete concepts we as humans understand and can use one or two words to describe, all from unstructured sensory data. This is very much like a human child, learning to recognize the concepts of “cat” and “horse” from the spoken feedback he or she receives from their mother. The question now becomes: can we learn to better discretize this real-valued high level semantic representation?

Currently, the four basic tools we have at our disposal for discretization are clustering, binning, argmax-ing, and sampling discrete distributions. Each has its pros and cons, and there are many new directions of research with the goal of unifying these non-differentiable approaches (often performed after training) with stochastic gradient descent to shape low-level weights of the neural network at training time. None has taken off as “*the* way” to learn a discrete variable in a neural architecture. It may be one of these approaches, or it may be an entirely different approach for going from continuous to discrete and optimizing through that transformation. However, once a method is found, it may very well be possible to learn a latent sequence of discrete variables representing a learned written language, learned in an entirely unsupervised fashion from speech and other sensory input. In this learned written language, it will be possible to use grammars, use N-grams, learn autoregressive distributions, and do everything one would do with text instead with the discrete representation.

I conclude with a reflection. My contributions in this thesis are a small step in the direction of a lofty goal: learning to understand spoken language in an unsupervised fashion. For the goal to be realized, it will require collaboration from researchers across labs, across disciplines even. The following is certain: it’s an exciting time for machine learning and artificial intelligence.

Appendix A

Model Architectures

In this Appendix, I give a description of the different DAVEnet encoder modules used in this thesis. Note that for the TIDIGITS/MNIST experiments in Chapter 3, the model descriptions are given in Table 3.1. In this section, the embedding dimension (typically 1024) is referred to as D. For convolutions, constant padding before the convolution is used, as is standard with PyTorch (Paszke et al. 2017). All models in this section are implemented in PyTorch (Paszke et al. 2017).

A.1 Traditional DAVEnet Architecture

First the image encoder:

1. VGG16 through Conv 5-3 (final maxpool and fully connected layers removed) (Simonyan and Zisserman 2014)
2. 3×3 convolution with stride 1 and padding 1, from 512 channels to D channels

the output is downsampled spatially by a factor of 16 as compared to the input.

Next the audio encoder. The input is a $40 \times T$ mel-spectrogram with 1 channel.

1. BatchNorm 2D (Ioffe and Szegedy 2015)
2. 40×1 convolution with stride 1, padding 0, from 1 channel to 128 channels
3. 1×11 convolution with stride 1, padding $(0, 5)$, from 128 channels to 256 channels
4. 1×3 max pooling with stride $(1, 2)$, padding $(0, 1)$
5. 1×17 convolution with stride 1, padding $(0, 8)$, from 256 channels to 512 channels
6. 1×3 max pooling with stride $(1, 2)$, padding $(0, 1)$
7. 1×17 convolution with stride 1, padding $(0, 8)$, from 512 channels to 512 channels
8. 1×3 max pooling with stride $(1, 2)$, padding $(0, 1)$
9. 1×17 convolution with stride 1, padding $(0, 8)$, from 512 channels to D channels

the output is downsampled temporally by a factor of 8 as compared to the input.

A.2 Residual DAVEnet Architecture

First the image encoder:

1. ResNet-50 with final average pool and fully connected layer removed (He et al. 2016)
2. 1×1 convolution with stride 1 and padding 0, from 2048 channels to D channels

the output is downsampled spatially by a factor of 32 as compared to the input.

Next the audio encoder: the residual audio encoder is more complicated to describe than the models listed thus far. It consists of a series of blocks of the form:

1. 1×9 convolution with stride (1, stride), padding $(0, 4)$, from D_{in} channels to D_{planes} channels
2. BatchNorm 2D (Ioffe and Szegedy 2015)
3. ReLU
4. 1×9 convolution with stride 1, padding $(0, 4)$, from D_{planes} channels to D_{planes} channels
5. BatchNorm 2D (Ioffe and Szegedy 2015)

The output of this block is added to a downsampled residual connection of the input to the block. The input is downsampled using:

1. 1×1 convolution with stride (1, stride), padding 1, from D_{in} channels to D_{planes} channels
2. BatchNorm 2D (Ioffe and Szegedy 2015)

if the stride for the block is not 1. If the stride for the block is 1, the input itself is used as the residual connection. We refer to this residual block architecture simply as "Residual block" in the following model description. Note that the input is a $40 \times T$ mel-spectrogram with 1 channel.

1. 40×1 convolution with stride 1, padding $(0, 1)$, no bias, from 1 channel to 128 channels.
2. BatchNorm 2D (Ioffe and Szegedy 2015)
3. ReLU
4. Residual block from 128 channels to 128 channels with stride 2
5. ReLU
6. Residual block from 128 channels to 128 channels with stride 1
7. ReLU
8. Residual block from 128 channels to 256 channels with stride 2
9. ReLU
10. Residual block from 256 channels to 256 channels with stride 1
11. ReLU
12. Residual block from 256 channels to 512 channels with stride 2
13. ReLU
14. Residual block from 512 channels to 512 channels with stride 1
15. ReLU
16. Residual block from 512 channels to D channels with stride 2
17. ReLU
18. Residual block from D channels to D channels with stride 1
19. ReLU

the output is downsampled temporally by a factor of 8 as compared to the input.

Appendix B

Word Detectors

In Chapter 4, I describe a series of experiments I ran training a DAVEnet model on the new Captioned Moments dataset. At the end, I take the most successful DAVEnet, which took 3 video frames and the ambient audio as input, and used network dissection (Bau et al. 2017; Zhou et al. 2017) to determine how training on the Captioned Moments dataset affected the model's word detection and understanding. I argued that the model's increased ability to detect action words, such as “running” and “talking”, showed that training on the Captioned Moments dataset might enable DAVEnet to learn more action-related concepts.

In this Appendix, I show the 25 dimensions with the word detectors with greatest IOU scores for both the Places model and the fine-tuned Captioned Moments model. Then I give the full version of Table 4.8.

From the Places model:

```
Dim. 529  trees (0.509), tree (0.064)
Dim. 363  trees (0.407)
Dim. 92   red (0.405), orange (0.052)
Dim. 391  trees (0.372), tree (0.122)
Dim. 172  water (0.340), ocean (0.096), river (0.091), beach (0.058)
Dim. 994  people (0.337), walking (0.058)
Dim. 53   snow (0.325)
Dim. 890  white (0.321)
Dim. 29   building (0.285), buildings (0.131)
Dim. 838  table (0.282), tables (0.157), chairs (0.082)
Dim. 751  black (0.273)
Dim. 707  room (0.273), inside (0.061)
Dim. 962  blue (0.270)
Dim. 601  white (0.269)
Dim. 598  green (0.249), grass (0.051)
Dim. 995  man (0.247), men (0.079)
Dim. 278  yellow (0.246), green (0.066)
Dim. 463  field (0.228), grass (0.195)
Dim. 249  sitting (0.227)
Dim. 131  building (0.221)
Dim. 710  walking (0.220)
Dim. 801  sitting (0.211), standing (0.109)
Dim. 579  black (0.205), white (0.057)
Dim. 467  red (0.196)
Dim. 419  people (0.193), baseball (0.059)
```

From the fine-tuned model:

```
Dim. 529  trees (0.420), tree (0.073)
Dim. 172  water (0.357), ocean (0.081), beach (0.054)
Dim. 391  trees (0.344), tree (0.139)
Dim. 92   red (0.330)
Dim. 29   building (0.325), buildings (0.110)
Dim. 363  trees (0.322), grass (0.093)
Dim. 53   snow (0.307)
Dim. 994  people (0.305)
Dim. 890  white (0.303)
Dim. 838  table (0.276), tables (0.114), chairs (0.084)
Dim. 1116 people (0.266)
Dim. 598  green (0.258)
Dim. 751  black (0.255)
Dim. 579  black (0.244), white (0.058)
Dim. 995  man (0.241), men (0.051)
Dim. 601  white (0.238), wooden (0.050)
Dim. 278  yellow (0.228), orange (0.054)
Dim. 396  trees (0.226), tree (0.110)
Dim. 962  blue (0.218)
Dim. 903  building (0.211), buildings (0.075), house (0.072)
Dim. 463  field (0.205), grass (0.193)
Dim. 158  bridge (0.195)
Dim. 239  inside (0.194)
Dim. 60   sky (0.191), skies (0.053)
Dim. 815  building (0.190), buildings (0.084)
```

The next table shows words, sorted from those which moved forward the most in the sorted list of max IOU-ranking when fine tuned on Captioned Moments to those that moved backwards the most. Simply put, the list is sorted by words which were learned through fine-tuning to words that were forgotten. Green bold words correspond to "learned" words and blue words to "remembered" words. Red italicized words correspond to "forgotten" words. Only the 160 words with IOU greater than 0.04 in either the Places model or the Captioned Moments model were considered. Words with IOU greater than 0.05 were considered known for the purpose of deciding which words were forgotten, retained, or learned.

The parenthesized integer next to the IOU score corresponds to the dimension of the embedding with that particular IOU score. Dimensions greater than 1024 are shown in red and correspond to ambient video audio dimensions. Notice that "talking", "dog", and "machine" have detectors in dimensions 1128 and 1119 (ambient video audio dimensions) of the embedding.

Word	**IOU Before**	**IOU After**	**Rank Before**	**Rank After**	**# Passed**	**# Passed By**
talking	0.037 (1)	0.104 (**1128**)	145 / 160	48 / 160	97	0
hands	0.028 (1022)	0.080 (937)	157 / 160	73 / 160	84	0
baby	0.073 (22)	0.163 (876)	84 / 160	25 / 160	59	0
wood	0.041 (1023)	0.069 (1023)	139 / 160	85 / 160	56	2
bicycle	0.036 (866)	0.057 (866)	149 / 160	98 / 160	52	1
ball	0.032 (277)	0.054 (49)	152 / 160	104 / 160	49	1
dog	0.047 (276)	0.077 (**1119**)	122 / 160	76 / 160	48	2
camera	0.048 (1004)	0.077 (822)	121 / 160	77 / 160	47	3
game	0.037 (749)	0.055 (97)	146 / 160	103 / 160	45	2
large	0.078 (691)	0.121 (691)	78 / 160	38 / 160	41	1
he	0.030 (188)	0.052 (188)	154 / 160	115 / 160	40	1
running	0.036 (301)	0.053 (64)	148 / 160	110 / 160	41	3
she	0.028 (1002)	0.049 (282)	156 / 160	119 / 160	38	1
wearing	0.042 (187)	0.056 (16)	136 / 160	99 / 160	41	4
two	0.037 (231)	0.053 (542)	147 / 160	111 / 160	38	4
dark	0.060 (513)	0.096 (606)	91 / 160	57 / 160	35	1
outside	0.081 (650)	0.120 (650)	73 / 160	39 / 160	36	2
young	0.082 (306)	0.131 (614)	71 / 160	37 / 160	35	1
kids	0.028 (154)	0.046 (154)	158 / 160	125 / 160	33	0
girls	0.043 (856)	0.056 (856)	132 / 160	100 / 160	37	5
group	0.046 (813)	0.060 (813)	125 / 160	94 / 160	34	3
picture	0.046 (854)	0.061 (230)	124 / 160	93 / 160	34	3
car	0.097 (418)	0.150 (418)	54 / 160	27 / 160	28	1
cartoon	0.026 (259)	0.043 (980)	159 / 160	133 / 160	26	0
flower	0.032 (155)	0.046 (329)	153 / 160	127 / 160	30	4
bushes	0.051 (839)	0.063 (720)	115 / 160	90 / 160	30	5
girl	0.074 (856)	0.094 (856)	83 / 160	58 / 160	28	3
watching	0.044 (64)	0.053 (225)	130 / 160	106 / 160	32	8
cars	0.080 (418)	0.103 (418)	74 / 160	50 / 160	27	3
person	0.087 (492)	0.112 (950)	66 / 160	42 / 160	28	4
hair	0.024 (1004)	0.041 (1004)	160 / 160	137 / 160	23	0
someone	0.029 (966)	0.044 (492)	155 / 160	132 / 160	26	3
says	0.042 (167)	0.053 (199)	134 / 160	112 / 160	31	8
child	0.051 (306)	0.066 (306)	110 / 160	87 / 160	28	5
bed	0.040 (375)	0.048 (501)	143 / 160	122 / 160	30	9
waterfall	0.042 (791)	0.050 (928)	138 / 160	117 / 160	30	9
stage	0.087 (105)	0.105 (97)	64 / 160	47 / 160	23	5
dressed	0.057 (522)	0.074 (522)	97 / 160	80 / 160	21	4
shop	0.035 (412)	0.041 (412)	150 / 160	136 / 160	22	8
structure	0.042 (631)	0.048 (384)	135 / 160	121 / 160	26	12
football	0.033 (503)	0.041 (97)	151 / 160	138 / 160	22	9
boy	0.054 (306)	0.062 (306)	103 / 160	91 / 160	19	7
couch	0.040 (99)	0.045 (43)	140 / 160	130 / 160	22	12
baseball	0.079 (49)	0.087 (49)	77 / 160	67 / 160	15	5
floor	0.119 (915)	0.140 (915)	42 / 160	32 / 160	12	2
shirt	0.123 (811)	0.146 (811)	39 / 160	29 / 160	12	2
night	0.070 (984)	0.075 (984)	87 / 160	79 / 160	13	5
beach	0.074 (319)	0.078 (319)	82 / 160	75 / 160	12	5
tree	0.122 (391)	0.139 (391)	40 / 160	33 / 160	10	3
inside	0.175 (239)	0.194 (239)	23 / 160	16 / 160	7	0
tractor	0.042 (701)	0.044 (701)	137 / 160	131 / 160	21	15
forest	0.039 (485)	0.041 (936)	144 / 160	139 / 160	21	16
skies	0.051 (60)	0.053 (60)	114 / 160	109 / 160	19	14
flowers	0.071 (213)	0.074 (213)	86 / 160	81 / 160	12	7
ground	0.090 (466)	0.097 (466)	61 / 160	56 / 160	13	8
street	0.134 (30)	0.149 (30)	33 / 160	28 / 160	7	2
wooden	0.165 (1023)	0.186 (1023)	24 / 160	19 / 160	5	0
bridge	0.193 (158)	0.195 (158)	20 / 160	15 / 160	5	0
orange	0.052 (92)	0.054 (278)	109 / 160	105 / 160	18	14
jacket	0.055 (624)	0.059 (624)	100 / 160	96 / 160	14	10
men	0.083 (717)	0.088 (717)	69 / 160	65 / 160	12	8
woman	0.160 (74)	0.179 (74)	26 / 160	22 / 160	4	0
shorts	0.047 (811)	0.049 (811)	123 / 160	120 / 160	20	17
plants	0.059 (720)	0.063 (720)	92 / 160	89 / 160	10	7
green	0.249 (598)	0.258 (598)	12 / 160	9 / 160	3	0

Word	IOU Before	IOU After	Rank Before	Rank After	# Passed	# Passed By
building	0.285 (29)	0.325 (29)	7 / 160	4 / 160	3	0
yellow	0.246 (278)	0.228 (278)	14 / 160	12 / 160	2	0
man	0.247 (995)	0.241 (995)	13 / 160	11 / 160	2	0
sky	0.193 (60)	0.191 (60)	19 / 160	18 / 160	3	2
grass	0.195 (463)	0.193 (463)	18 / 160	17 / 160	3	2
field	0.228 (463)	0.205 (463)	15 / 160	14 / 160	1	0
water	0.340 (172)	0.357 (172)	3 / 160	2 / 160	1	0
chair	0.040 (637)	0.040 (637)	142 / 160	142 / 160	18	18
table	0.282 (838)	0.276 (838)	8 / 160	8 / 160	0	0
snow	0.325 (53)	0.307 (53)	5 / 160	5 / 160	1	1
trees	0.509 (529)	0.420 (529)	1 / 160	1 / 160	0	0
fire	0.074 (77)	0.072 (77)	81 / 160	82 / 160	10	11
black	0.273 (751)	0.255 (751)	9 / 160	10 / 160	0	1
white	0.321 (890)	0.303 (890)	6 / 160	7 / 160	0	1
red	0.405 (92)	0.330 (92)	2 / 160	3 / 160	0	1
statue	0.045 (665)	0.045 (665)	126 / 160	128 / 160	17	19
church	0.084 (384)	0.084 (305)	68 / 160	70 / 160	8	10
top	0.093 (1)	0.093 (1)	58 / 160	60 / 160	9	11
boxing	0.098 (639)	0.100 (639)	53 / 160	55 / 160	7	9
rock	0.104 (688)	0.102 (688)	51 / 160	53 / 160	7	9
wall	0.147 (847)	0.141 (847)	29 / 160	31 / 160	2	4
mountains	0.175 (763)	0.164 (763)	22 / 160	24 / 160	1	3
blue	0.270 (962)	0.218 (962)	11 / 160	13 / 160	1	3
people	0.337 (994)	0.305 (994)	4 / 160	6 / 160	0	2
rocks	0.092 (688)	0.090 (688)	59 / 160	63 / 160	7	11
background	0.136 (556)	0.134 (556)	31 / 160	35 / 160	2	6
sitting	0.227 (249)	0.183 (249)	16 / 160	20 / 160	1	5
snowy	0.045 (53)	0.042 (53)	129 / 160	134 / 160	18	23
stone	0.164 (384)	0.144 (384)	25 / 160	30 / 160	0	5
house	0.183 (849)	0.155 (849)	21 / 160	26 / 160	0	5
parking	0.058 (617)	0.056 (617)	95 / 160	101 / 160	9	15
playing	0.088 (62)	0.084 (368)	63 / 160	69 / 160	6	12
door	0.097 (248)	0.093 (248)	55 / 160	61 / 160	6	12
brick	0.116 (640)	0.104 (640)	43 / 160	49 / 160	2	8
road	0.124 (341)	0.110 (341)	38 / 160	44 / 160	3	9
mountain	0.141 (763)	0.133 (763)	30 / 160	36 / 160	1	7
river	0.148 (951)	0.135 (951)	28 / 160	34 / 160	1	7
walking	0.220 (710)	0.167 (710)	17 / 160	23 / 160	0	6
train	0.131 (333)	0.111 (333)	36 / 160	43 / 160	2	9
boat	0.087 (880)	0.081 (880)	65 / 160	72 / 160	5	13
standing	0.110 (822)	0.101 (653)	46 / 160	54 / 160	2	10
clouds	0.077 (431)	0.064 (431)	79 / 160	88 / 160	5	14
store	0.136 (212)	0.114 (212)	32 / 160	41 / 160	0	9
rocky	0.072 (688)	0.059 (688)	85 / 160	95 / 160	5	15
children	0.122 (306)	0.103 (616)	41 / 160	51 / 160	1	11
small	0.131 (942)	0.109 (942)	35 / 160	46 / 160	0	11
buildings	0.131 (29)	0.110 (29)	34 / 160	45 / 160	0	11
room	0.273 (707)	0.179 (707)	10 / 160	21 / 160	0	11
parked	0.043 (617)	0.039 (617)	133 / 160	145 / 160	14	26
pink	0.049 (712)	0.045 (712)	117 / 160	129 / 160	13	25
tower	0.051 (903)	0.047 (432)	111 / 160	123 / 160	12	24
tracks	0.057 (701)	0.053 (701)	96 / 160	108 / 160	7	19
classroom	0.043 (734)	0.039 (707)	131 / 160	144 / 160	14	27
restaurant	0.051 (233)	0.046 (735)	113 / 160	126 / 160	12	25
glass	0.082 (891)	0.070 (891)	70 / 160	83 / 160	4	17
dancing	0.105 (789)	0.090 (789)	49 / 160	62 / 160	3	16
tables	0.157 (838)	0.114 (838)	27 / 160	40 / 160	0	13
front	0.040 (204)	0.031 (510)	141 / 160	155 / 160	5	19
garden	0.054 (110)	0.050 (720)	104 / 160	118 / 160	8	22
desk	0.081 (870)	0.068 (736)	72 / 160	86 / 160	3	17
pool	0.091 (628)	0.080 (628)	60 / 160	74 / 160	3	17
window	0.112 (908)	0.093 (908)	45 / 160	59 / 160	1	15
ocean	0.096 (172)	0.081 (172)	56 / 160	71 / 160	2	17
chairs	0.129 (988)	0.102 (988)	37 / 160	52 / 160	0	15
fence	0.056 (525)	0.052 (167)	98 / 160	114 / 160	6	22

Word	IOU Before	IOU After	Rank Before	Rank After	# Passed	# Passed By
dirt	0.086 (513)	0.069 (345)	67 / 160	84 / 160	3	20
track	0.105 (701)	0.087 (701)	50 / 160	68 / 160	1	19
brown	0.105 (810)	0.088 (810)	48 / 160	66 / 160	1	19
shoes	0.045 (282)	0.039 (282)	128 / 160	146 / 160	12	31
machine	0.059 (474)	0.053 (**1050**)	93 / 160	113 / 160	5	24
kitchen	0.113 (137)	0.089 (506)	44 / 160	64 / 160	0	20
area	0.076 (844)	0.055 (844)	80 / 160	102 / 160	3	25
walls	0.080 (847)	0.059 (847)	75 / 160	97 / 160	2	24
stairs	0.055 (246)	0.047 (246)	101 / 160	124 / 160	6	29
city	0.103 (30)	0.076 (30)	52 / 160	78 / 160	1	27
cloudy	0.045 (431)	0.030 (155)	127 / 160	156 / 160	4	32
office	0.052 (147)	0.042 (707)	107 / 160	135 / 160	9	37
truck	0.048 (701)	0.036 (940)	120 / 160	151 / 160	7	38
lights	0.050 (984)	0.038 (638)	116 / 160	149 / 160	6	39
fountain	0.079 (791)	0.051 (487)	76 / 160	116 / 160	1	41
old	0.048 (384)	0.025 (384)	119 / 160	160 / 160	0	41
photograph	0.048 (186)	0.027 (233)	118 / 160	159 / 160	0	41
station	0.056 (438)	0.041 (800)	99 / 160	140 / 160	5	46
distance	0.088 (429)	0.053 (711)	62 / 160	107 / 160	1	46
windows	0.108 (908)	0.062 (908)	47 / 160	92 / 160	0	45
video	0.051 (1011)	0.029 (107)	112 / 160	158 / 160	0	46
market	0.052 (790)	0.035 (412)	108 / 160	154 / 160	1	47
painting	0.053 (935)	0.035 (825)	106 / 160	153 / 160	1	48
photo	0.054 (107)	0.037 (107)	102 / 160	150 / 160	1	49
around	0.054 (834)	0.030 (934)	105 / 160	157 / 160	0	52
side	0.068 (199)	0.040 (650)	89 / 160	141 / 160	2	54
tall	0.067 (706)	0.039 (355)	90 / 160	143 / 160	2	55
dining	0.058 (578)	0.035 (595)	94 / 160	152 / 160	0	58
castle	0.069 (477)	0.039 (384)	88 / 160	147 / 160	1	60
sign	0.093 (199)	0.039 (199)	57 / 160	148 / 160	0	91

Bibliography

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. In *Advances in Neural Information Processing Systems*, pages 892–900, 2016.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. *CoRR*, abs/1704.05796, 2017. URL `http://arxiv.org/abs/1704.05796`.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

Yu-An Chung and James R. Glass. Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech. *CoRR*, abs/1803.08976, 2018. URL `http://arxiv.org/abs/1803.08976`.

Herbert H. Clark and Susan E. Brennan. Grounding in communication. In Lauren Resnick, Levine B., M. John, Stephanie Teasley, and D., editors, *Perspectives on Socially Shared Cognition*, pages 13–1991. American Psychological Association, 1991.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 1991. ISBN 0471062596.

FFmpeg Developers. Ffmpeg tool. 2017. URL `http://ffmpeg.org/`.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE, 2013.

David Harwath and James R. Glass. Learning word-like units from joint audio-visual analysis. *Proceedings of the 2017 meeting of the Association for Computational Lingustics*, 2017. URL `http://aclweb.org/anthology/P17-1047`.

David Harwath, Antonio Torralba, and James Glass. Unsupervised learning of spoken language with visual context. In *Advances in Neural Information Processing Systems*, pages 1858–1866, 2016.

David Harwath, Adrià Recasens, Dídac Surís, Galen Chuang, Antonio Torralba, and James R. Glass. Jointly discovering visual objects and spoken words from raw sensory input. *CoRR*, abs/1804.01452, 2018a. URL `http://arxiv.org/abs/1804.01452`.

David F. Harwath, Galen Chuang, and James R. Glass. Vision as an interlingua: Learning multilingual semantic embeddings of untranscribed speech. *CoRR*, abs/1804.03052, 2018b. URL `http://arxiv.org/abs/1804.03052`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Wei-Ning Hsu, Yu Zhang, and James Glass. Learning latent representations for speech generation and transformation. *arXiv preprint arXiv:1704.04222*, 2017a.

Wei-Ning Hsu, Yu Zhang, and James R. Glass. Unsupervised learning of disentangled and interpretable representations from sequential data. *CoRR*, abs/1709.07902, 2017b. URL `http://arxiv.org/abs/1709.07902`.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. URL `http://arxiv.org/abs/1502.03167`.

Aren Jansen, Manoj Plakal, Ratheet Pandya, Daniel P. W. Ellis, Shawn Hershey, Jiayang Liu, R. Channing Moore, and Rif A. Saurous. Unsupervised learning of semantic audio representations. *CoRR*, abs/1711.02209, 2017. URL `http://arxiv.org/abs/1711.02209`.

Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

Karan Kashyap. Learning digits via joint audio-visual representations. Master's thesis, Massachusetts Institute of Technology, 2017.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL `http://arxiv.org/abs/1412.6980`.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). URL `http://www.cs.toronto.edu/~kriz/cifar.html`.

Q. V. Le. Building high-level features using large scale unsupervised learning. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8595–8598, May 2013. doi: 10.1109/ICASSP.2013.6639343.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE, 86(11):227-2324*, November 1998.

Chia-ying Lee and James Glass. A nonparametric bayesian approach to acoustic model discovery. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 40–49. Association for Computational Linguistics, 2012.

Kenneth Leidal, David Harwath, and James R. Glass. Learning modality-invariant representations for speech and images. *CoRR*, abs/1712.03897, 2017. URL `http://arxiv.org/abs/1712.03897`.

R Gary Leonard and George Doddington. Tidigits speech corpus. *Texas Instruments, Inc*, 1993.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

Kingson Man, Jonas T Kaplan, Antonio Damasio, and Kaspar Meyer. Sight and sound converge to form modality-invariant representations in temporoparietal cortex. *Journal of Neuroscience*, 32(47):16629–16636, 2012.

Mathew Monfort, Bolei Zhou, Sarah Adel Bargal, Alex Andonian, Tom Yan, Kandan Ramakrishnan, Lisa M. Brown, Quanfu Fan, Dan Gutfruend, Carl Vondrick, and Aude Oliva. Moments in time dataset: one million videos for event understanding. *CoRR*, abs/1801.03150, 2018. URL `http://arxiv.org/abs/1801.03150`.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Stavros Petridis, Themos Stafylakis, Pingchuan Ma, Feipeng Cai, Georgios Tzimiropoulos, and Maja Pantic. End-to-end audiovisual speech recognition. *CoRR*, abs/1802.06424, 2018. URL `http://arxiv.org/abs/1802.06424`.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

Nikhil Rasiwasia, Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert R.G. Lanckriet, Roger Levy, and Nuno Vasconcelos. A new approach to cross-modal multimedia retrieval. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 251–260, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-933-6. doi: 10.1145/1873951.1873987. URL `http://doi.acm.org/10.1145/1873951.1873987`.

Kuniaki Saito, Yusuke Mukuta, Yoshitaka Ushiku, and Tatsuya Harada. Demian: Deep modality invariant adversarial network. *arXiv preprint arXiv:1612.07976*, 2016.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Felix Sun, David Harwath, and James Glass. Look, listen, and decode: Multimodal speech recognition with images. In *Spoken Language Technology Workshop (SLT), 2016 IEEE*, pages 573–578. IEEE, 2016.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. *arXiv preprint arXiv:1702.05464*, 2017.

Kaiye Wang, Qiyue Yin, Wei Wang, Shu Wu, and Liang Wang. A comprehensive survey on cross-modal retrieval. *arXiv preprint arXiv:1607.06215*, 2016.

Yu Zhang and James Glass. Unsupervised spoken keyword spotting via segmental dtw on gaussian posteriorgrams. *Proc. ASRU, Merano, Italy*, December 2009.

Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ADE20K dataset. *CoRR*, abs/1608.05442, 2016. URL `http://arxiv.org/abs/1608.05442`.

Bolei Zhou, David Bau, Aude Oliva, and Antonio Torralba. Interpreting deep visual representations via network dissection. *CoRR*, abs/1711.05611, 2017. URL `http://arxiv.org/abs/1711.05611`.