Tampereen yliopisto

Paavo Pesä

CROSS-MODAL SEMANTIC RETRIEVAL IN NEURAL MODELS OF VISUALLY GROUNDED SPEECH

Faculty of Information Technology and Communication Sciences
Bachelor's thesis
February 2023

ABSTRACT

Paavo Pesä: Cross-modal semantic retrieval in neural models of visually grounded speech
Bachelor's thesis
Tampere University
Bachelor's Programme in Computing and Electrical Engineering
February 2023

Machine learning and its subset deep learning have been hot topics of technology for some years now. Visually grounded speech (VGS) models belong to a family of weakly supervised deep learning methods. The objective of VGS models is to learn to recognize semantic similarities between image and speech modalities without ever being trained with strong, ground-truth labels for spoken words and visual objects. Instead, training is done by only knowing the connection between the two modalities in the coarse level of images and spoken utterances. This form of weakly supervised learning is similar to what infants are experiencing while learning their native language.

The first part of this thesis discusses the visually grounded speech models, their usual structure, and the methods used while constructing them. In the latter part, this work focuses on the convolutional neural network (CNN) based VGS models and explores how the structure of speech encoder block can affect the performance of these models on semantic retrieval tasks. More specifically, I design and test three different variations of speech encoder block and compare the performance to the literature. My own implementations differ from the literature by: (i) lower receptive fields, (ii) increased number of filter channels, and (iii) higher receptive fields combined with the increased number of filter channels.

The results of this work show that modifying the hyperparameters of CNN audio encoders affects the VGS model performance. While decreasing the temporal receptive fields of the convolutional layers has a negative effect on cross-modal retrieval scores, increasing the receptive fields yields a better result in the cost of higher computational complexity and training time than the literature.

Keywords: cross-modal learning, neural network, deep learning, speech processing, visually grounded speech

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

TIIVISTELMÄ

Paavo Pesä: Multimodaalinen semanttinen haku visuaaliseen informaatioon kytketyn puheen neuromalleissa
Kandidaatin tutkielma
Tampereen yliopisto
Tieto- ja sähkötekniikan kandidaattiohjelma
Helmikuu 2023

Koneoppiminen ja sen osa-alue syväoppiminen ovat jo pidempään olleet suosittuja puheenaiheita teknologia-alalla. Visuaaliseen informaation kytketyn puheen mallit (engl. visually grounded speech (VGS)) kuuluvat niin kutsuttuihin heikosti valvottuihin syväoppimisen metodeihin (engl. weakly supervised learning). VGS-mallien tavoitteena on oppia tunnistamaan semanttisia samankaltaisuuksia kuvan ja puheen modaliteettien välillä ilman tarkkaan luokiteltua opetusdataa puheen sanoista ja kuvien visuaalisista objekteista. Sen sijaan kyseiset mallit opetetaan ainoastaan tiedolla siitä, mikä kuva ja puhuttu kuvaus liittyvät toisiinsa. Tämän tyylinen heikosti valvottu oppiminen on samankaltaista, mitä vauvat kokevat oppiessaan äidinkieltään.

Tämän työn ensimmäinen osa käsittelee visuaaliseen informaatioon kytketyn puheen malleja, niiden tyypillistä rakennetta sekä metodeja, joita mallien rakentamiseen tarvitaan. Työn jälkimmäisessä osassa keskitytään konvoluutioneuroverkkoon (CNN) perustuviin VGS-malleihin ja tutkitaan kuinka puhe-enkooderin rakenne vaikuttaa mallien suorituskykyyn semanttisissa hakutehtävissä. Tarkemmin sanottuna, tässä työssä suunnittelen ja testaan kolmea eri puhe-enkooderi variaatiota sekä vertaan niiden suorituskykyä kirjallisuuteen. Kehittämäni toteutukset eroavat kirjallisuudesta seuraavin tavoin: (i) pienemmät reseptiiviset kentät, (ii) suurempi määrä suodattimia ja (iii) suuremmat reseptiiviset kentät yhdistettynä suurempaan suodatinmäärään.

Työn tulokset osoittavat, että CNN-audioenkooderin hyperparametrien säätäminen vaikuttaa VGS-mallien suorituskykyyn. Tuloksista käy myös ilmi, että konvoluutiokerrosten reseptiivisten kenttien pienentäminen vaikuttaa negatiivisesti multimodaalisiin hakutuloksiin. Sen sijaan reseptiivisten kenttien kasvattaminen tuottaa kirjallisuutta paremman tuloksen mallin laskennallisen monimutkaisuuden ja opetusajan kasvun kustannuksella.

Avainsanat: multimodaalinen oppiminen, neuroverkot, syväoppiminen, puheen prosessointi visuaaliseen informaatioon kytketty puhe

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

AF	activation function
AI	artificial intelligence
ANN	artificial neural network
CELL	cross-channel early lexical learning
CNN	convolutional neural network
DL	deep learning
DNN	deep neural network
FaST-VGS	fast-slow transformer for visually grounding speech
FC	fully connected
FNN	feedforward neural network
GPU	graphics processing unit
ML	machine learning
NN	neural network
RAM	random access memory
ReLU	rectified linear unit
RF	receptive field
RHN	recurrent highway network
RNN	recurrent neural network
VGS	visually grounded speech

1. INTRODUCTION

When babies learn a language, they rely on a wide range of indirect audio-visual clues [1]. They must learn to identify the objects in the world and the words that refer to them. The way babies learn to understand speech and recognize objects is in a very weakly supervised manner, not with the help of ground-truth notations, but by observation, repetition, and environmental interaction [1][2].

In this thesis, a type of deep neural network inspired by the way children acquire their first language skills, a visually grounded speech (VGS) model is presented. VGS models belong to deep learning methods where speech and image representations are learned without any explicit labels. Instead, these models are trained using correct mappings between images and spoken captions describing the images. The basic idea behind VGS models is to utilize the co-occurrences in audio-visual data. For instance, from the spoken captions “A dog is holding a frisbee in its mouth” and “A dog running on a field” aided with images of these scenes, a model could learn to link the audio signal for “dog” to the visual representation of a dog because it is common to both audio-visual pairs. [3]

A typical VGS model is structured as follows. Speech and image data are first processed in separate parallel channels and then mapped together to a shared semantic space through a similarity score. After discussing the typical VGS models and their fundamentals, this work focuses on the convolutional neural network (CNN) based VGS models. The goal is to optimize the speech encoder block and explore the effects on the model performance by tuning the hyperparameters in its layers.

1.1 Historical developments

Visually grounded speech models have been developed for the last twenty years. Several technological fields such as machine learning, speech processing, and computer vision have contributed to the development of VGS models. [1] One of the very first computational implementations of learning a spoken language with the help of visual grounding was a Cross-channel Early Lexical Learning (CELL) model [4]. The tasks of the CELL model were to segment speech into words, form visual categories from images, and finally, associate segmented words to the formed visual categories. The experiments with

the CELL model showed that it was feasible to learn word meanings with visual grounding. However, the study had some limitations such as the small-scale data used and the rather artificial way they collected visual data in a laboratory setting by capturing images one object at a time.

In [5], they further studied the importance of the speaker's gaze and pointing gestures as a source of information for children when learning a language. The biggest difference to the earlier CELL model work was the data collection process in which authors collected data not only from spoken utterances but also their eye gaze as well as head and hand position. In their work, they concluded that the model which uses speaker attention data performed better than the baseline model which ignored gaze and head position information.

What the initial VGS implementations lacked was the size of available datasets. The advances in deep learning and data collection methods helped to move on from small and private datasets to larger and publicly shared datasets. The two most important datasets used later in VGS models were Flickr8K [6] with 8 000 images and MSCOCO (Microsoft Common Objects in Context) [7] with 123 287 images. Both datasets consisted of photographs of everyday objects and five brief textual descriptions describing the contents of each image.

SPEECH-COCO dataset [8] further extended the MSCOCO dataset by generating speech captions with text-to-speech synthesis to the textual descriptions. Even further, the SpokenCOCO dataset [9] collected 742 hours of speech from 2 352 people reading the aforementioned captions aloud. This SpokenCOCO dataset aimed to provide more diverse and natural speech captions compared to the SPEECH-COCO dataset. An example from the SpokenCOCO dataset can be seen in figureFigure 1.1.

Figure 1.1. *An example image and five spoken captions as text transcripts from the SpokenCOCO dataset [9].*

The SPEECH-COCO and SpokenCOCO datasets helped to start the development of large-scale neural models for cross-modal learning between image and speech. In [10], they reported experiments with a model that learned multimodal embeddings of data. Their experiments used so-called triplet loss as a contrastive loss for the first time in VGS applications.

Multiple different audio encoder architectures have been developed and experimented with in VGS models in the last six years. Convolutional audio encoder architecture was experimented with first (e.g., [11]). In [12], they presented an encoder that consists of a convolutional layer followed by recurrent highway network (RHN) layers. More recent work (e.g., [13]) experimented further with purely recurrent neural network (RNN) based audio encoder architectures with an attention layer.

The latest research (e.g., [14]) experiments with a Transformer-based model for the first time in speech-image retrieval tasks. Their work presented a model called Fast-Slow Transformer for Visually Grounding Speech (FaST-VGS) which achieved state-of-the-art retrieval accuracy scores on benchmark datasets.

The rest of this thesis is structured as follows. The necessary background and methods used in constructing a VGS model are introduced in Chapter 2. Some variations of speech encoder blocks are introduced, and the experimental setup and results are discussed in Chapter 3. Finally, Chapter 4 summarizes this thesis and discusses the results gathered.

2. BACKGROUND

Machine learning (ML) has been one of the fastest-growing fields and hot topics of technology for many years now. ML is a subset of artificial intelligence (AI) that is used to teach machines how to handle data more efficiently [15]. Over recent years, the success and growth of ML are credited to the abundance of data in different forms (big data), and to the increased computational power available [16]. As the era of big data continues to grow there are no signs that the popularity of ML and AI will plateau any time soon [17].

There are a few different approaches when it comes to data used in machine learning applications. Supervised and unsupervised learning are the two most common branches of ML. Supervised learning is often used for classification and regression tasks whereas unsupervised learning is usually used for tasks such as segmenting data or clustering data between similar features [18]. Another difference between both approaches is the data type. Supervised learning utilizes high-cost ground-truth labeled data but in unsupervised learning annotated data is not needed. Thus, unsupervised algorithms can use raw data which is generally cheap and easily available.

ML applications use different algorithms to solve different data problems. A linear regression algorithm is an approach of supervised learning, where the simplest form is to fit a straight line to the dataset to determine the output label. Another popular ML algorithm is gradient descent where the objective is to minimize the cost function through an iterative process. [19]

2.1 Neural networks

One ML algorithm is a neural network (NN). A standard neural network or artificial neural network (ANN) consists of multiple connected processors called neurons which are inspired by and named after the structure of neurons in the human brain. Simple neural networks consist of 3 layers: input layer, hidden layer, and output layer. Figure Figure 2.1 illustrates the connections of neurons and the overall NN architecture.

Each neuron in a neural network connects to every neuron in the previous and the next layer. In addition, each neuron has an associated weight through which data goes from layer to layer. The weights are parameters of the model and are adjusted during the training stage. A neuron can be viewed as a computational unit that takes inputs and provides outputs. The formula for a single-layer neural network with one hidden layer can be shown as

$$H = XW^{(1)} + b^{(1)}, \tag{2.1}$$

where H is the output of the hidden layer, X denotes the inputs, $W^{(1)}$ is the hidden layer weights, and $b^{(1)}$ is the hidden layer bias. The result from the output layer can then be calculated as

$$O = HW^{(2)} + b^{(2)}, \tag{2.2}$$

Where O is the output of the output layer, $W^{(2)}$ is the weights of the output layer, and $b^{(2)}$ is the output layer bias. [20]

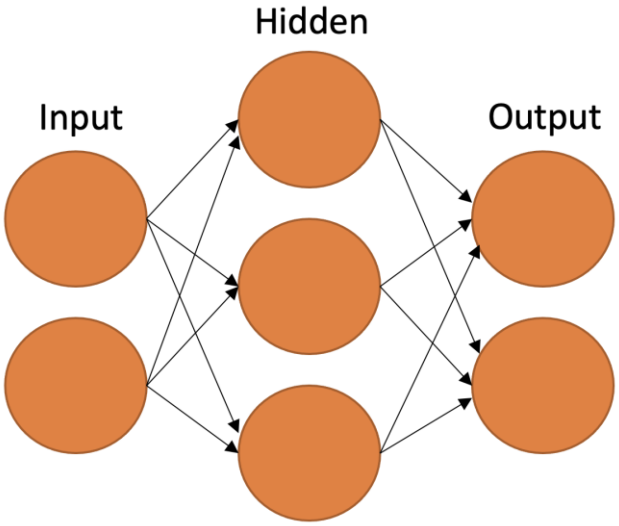

Figure 2.1. *A simple neural network*

To decide whether a neuron in a NN should be activated or not and to create nonlinearity, an activation function (AF) is needed. One commonly used activation function is rectified linear unit (ReLU) which has an easy-to-understand formula; output is zero for negative input and identity function, meaning input and output are equal, for positive input. Mathematically ReLU function can be expressed as

$$ReLU(x) = \max(0, x) = \begin{cases} x, & if\ x \geq 0 \\ 0, & otherwise \end{cases}. \tag{2.3}$$

Another popular activation function used in NNs is the sigmoid function also known as the logistic function and is given as

$$Logistic\ Sigmoid(x) = \frac{1}{1 + e^{-x}}. \tag{2.4}$$

A third commonly used function is the hyperbolic tangent function also known as the tanh function which is given as [21]

$$Tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} - e^{-x}}. \tag{2.5}$$

All forementioned activation functions are graphically illustrated in figure Figure 2.2.

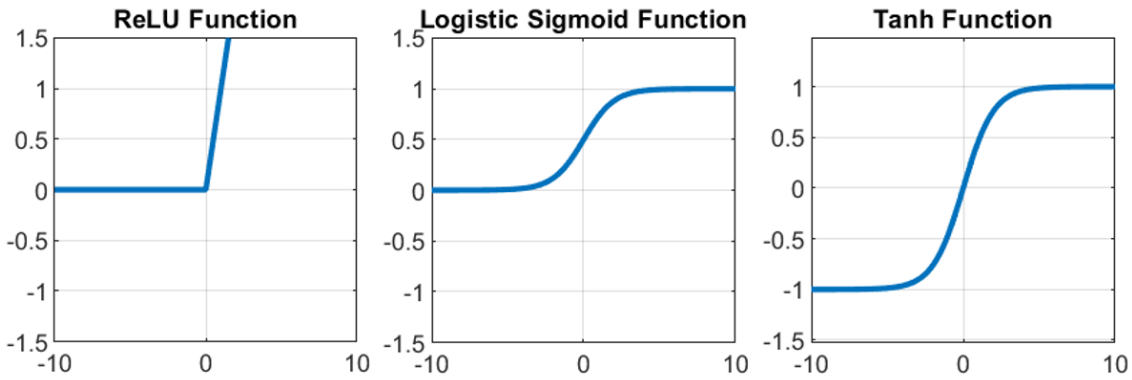

Figure 2.2. *An illustration of ReLU, Logistic Sigmoid and Tanh AFs.*

To train, validate, and test a neural network model, a dataset is divided into three separate sets of training, validation, and testing. The training set is used to adjust model parameters to solve a specific task. Training is done by running the full training dataset through a NN model multiple times. Once the model has processed a complete set of training data, it is said that one epoch is completed. Generally, training a neural network requires many epochs depending on the task and data. [20]

The main task of the validation set is to prevent model overfitting to training data. Meaning, the validation set is used to ensure that the model does not just improve on the data used in the training set, but also learns to generalize and use new data it has not seen before. Once the model has completed the training, the test dataset is used to test the model. The test set provides a final, unbiased metric on the model performance. The typical data split in ML applications is 80% of the whole dataset to the training set and 20% to test and validation sets. [20]

2.2 Deep learning

If a neural network consists of more than one hidden layer, it is usually referred to as a “Deep” neural network (DNN). The sheer amounts of layers and neurons in DNNs attempt to stimulate the human brain even better than the shallower NNs. In simple form, deep neural models can consist of a stack of “Feedforward” layers each followed by an activation function. In a feedforward neural network (FNN), the information flows in one direction from input to output. [22]

Deep learning (DL) is a subset of machine learning which refers to the models that make use of DNNs. While machine learning algorithms generally use structured data features to operate with, DL algorithms eliminate the need of some data preprocessing typically applied for ML models. In other words, in ML specific features are defined and extracted

from the input signal, while DL models can be applied directly to the raw signal. Deep learning models are widely used in a variety of applications such as computer vision, speech recognition, and self-driving cars. [20]

2.3 Convolutional neural networks

Convolutional neural networks are deep neural networks commonly used with image, speech, or audio input signals due to their superior performance in mentioned applications. CNN usually has three main types of layers: convolutional layer, pooling layer, and fully connected layer. A convolutional layer performs a mathematical operation called convolution which can be expressed as

$$s(t) = (x * w)(t), \tag{2.6}$$

where *x* is the input and *w* is the kernel. The resulting output is sometimes referred to as a feature map. [23]

As the name suggests, the convolutional layer is the main building block in CNN architecture where the majority of the computations occur. A kernel is a convolutional filter with a specific pre-defined size. The weights of the convolutional filter are usually initialized randomly but then trained and modified during the training process. The convolution operation goes as follows. The kernel and its weights are applied to an area of the input data and the result is fed to an output layer. Afterward, the filter shifts by a stride and repeats the aforementioned steps until the input data is fully processed.

There are in total four hyperparameters that affect the output dimensions of the convolutional layer: the number of filter channels, stride, kernel size, and padding size. The number of filter channels determines the depth of the output. Padding helps not to lose information from edges and corners. [23]

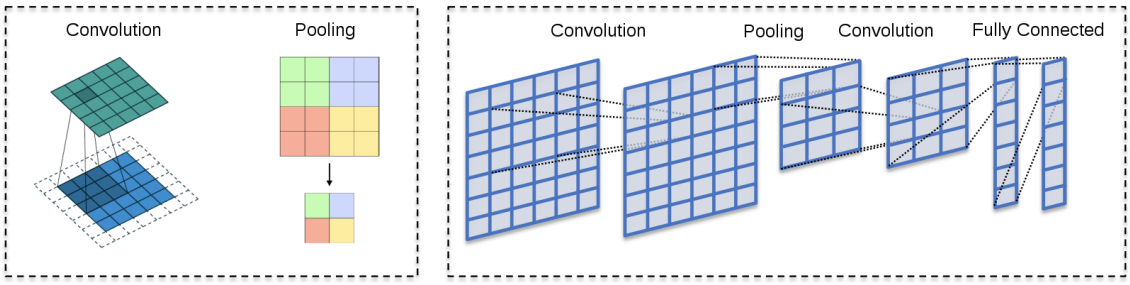

Figure 2.3. *Left: 3x3 convolution and 2x2 pooling examples. Right: Basic convolutional neural network architecture and the layers in it. [24]*

Usually, after each convolutional layer, there is a pooling layer, also known as downsampling, to reduce each feature map dimension. Maxpooling is one of the pooling types in

which the filter moves across the input and selects the maximum value as output. Typically, towards the end of a CNN architecture, fully connected (FC) layers are applied. FC layers use a flattened input where each input is connected to all neurons. [23] Figure Figure 2.3 illustrates the convolutional layer and pooling layer operations and shows an example of how convolutional and pooling layers are typically combined in an alternating manner to construct a CNN.

2.4 VGS Model

The usual structure of VGS models consists of a deep neural network with two branches, one responsible of image processing, and one of speech processing. The basic architecture of the VGS model is illustrated in figure Figure 2.4. The task of the VGS models is to learn the semantic similarities between image and speech modalities. The VGS models are trained through contrastive loss (see section 2.6) that tries to bring the corresponding image and speech samples close together and at the same time push the unrelated samples away from each other.

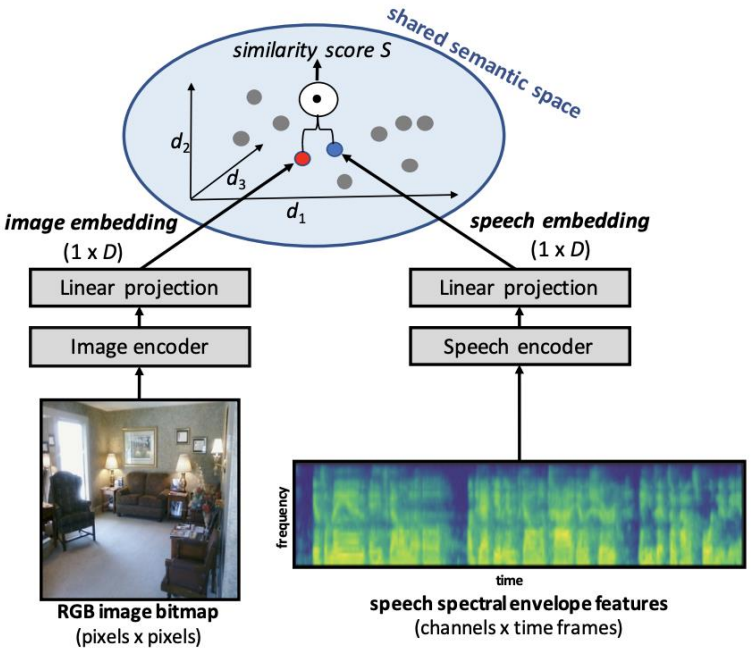

Figure 2.4. *VGS model structure [25].*

The image encoder takes the pixel-level input and outputs a high-level feature representation of image contents, while the speech encoder takes the speech signal, usually log Mel spectrum, as input and outputs high-level feature representation of audio contents. In common cases, both speech and image encoders consist of a stack of neural layers. Once through their respective branches, outputs are mapped together via a similarity score, such as dot product or cosine similarity, to form a semantic space.

2.5 Audio preprocessing

The mel scale describes the perceptual distance on how humans hear pitches of different frequencies. Mel scale can be mathematically calculated as

$$Mel(f) = 2595 log_{10}\left(1 + \frac{f}{700}\right), \tag{2.7}$$

where *f* is the frequency in hertz. [26]

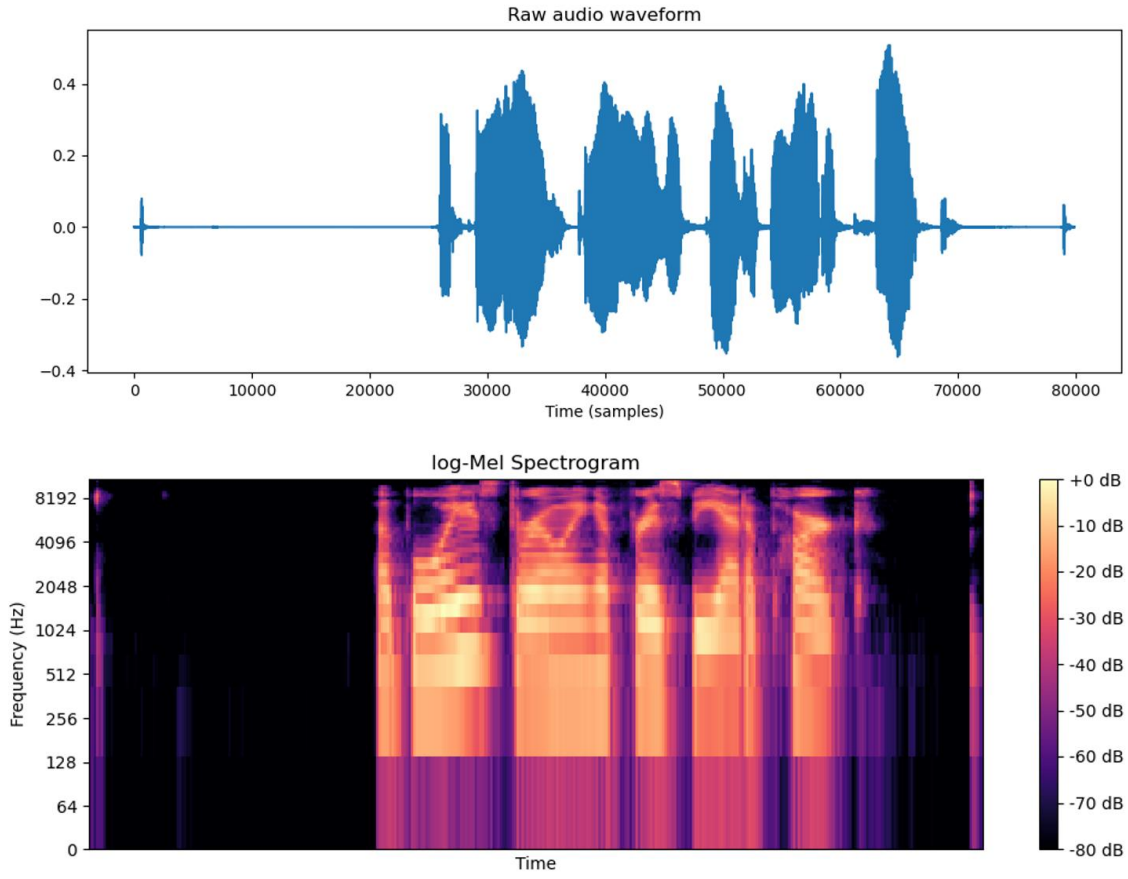

Figure 2.5. *Up: Raw audio waveform of a woman saying: "A boy is throwing a frisbee at a park." Down: log-mel spectrogram extracted from the same audio sample.*

Humans are good at recognizing differences in low audio frequencies and can easily tell a difference between 500 and 1 000 Hz. But the higher the frequencies get, the more distinguishable the differences for us humans' sound. Therefore, telling the difference between 10 000 and 10 500 Hz gets much harder. A spectrogram is a way to visually represent an audio signal's amplitude as it varies at different frequencies at different times [27]. To represent the spoken audio captions, log-mel filterbank spectrograms (also referred to as log-mel energies) are extracted from the audio files. This representation simulates the way the human ear picks up frequency variability. Figure Figure 2.5 illustrates the input speech waveform and a log-mel spectrogram extracted from it.

2.6 Contrastive loss

Contrastive learning is a machine learning technique used especially in unsupervised learning. The idea of contrastive learning is that samples are contrasted against each other to learn the shared features and attributes of samples. [28] To learn the cross-modal similarities, cross-modal contrastive learning is utilized as a training mechanism. In practice, samples that are similar to each other are mapped closer to each other in the semantic space, and dissimilar samples are pushed further apart. Figure Figure 2.6 illustrates how image and speech representations are mapped to the embedding space and how semantically similar features form clusters where distances are minimized.

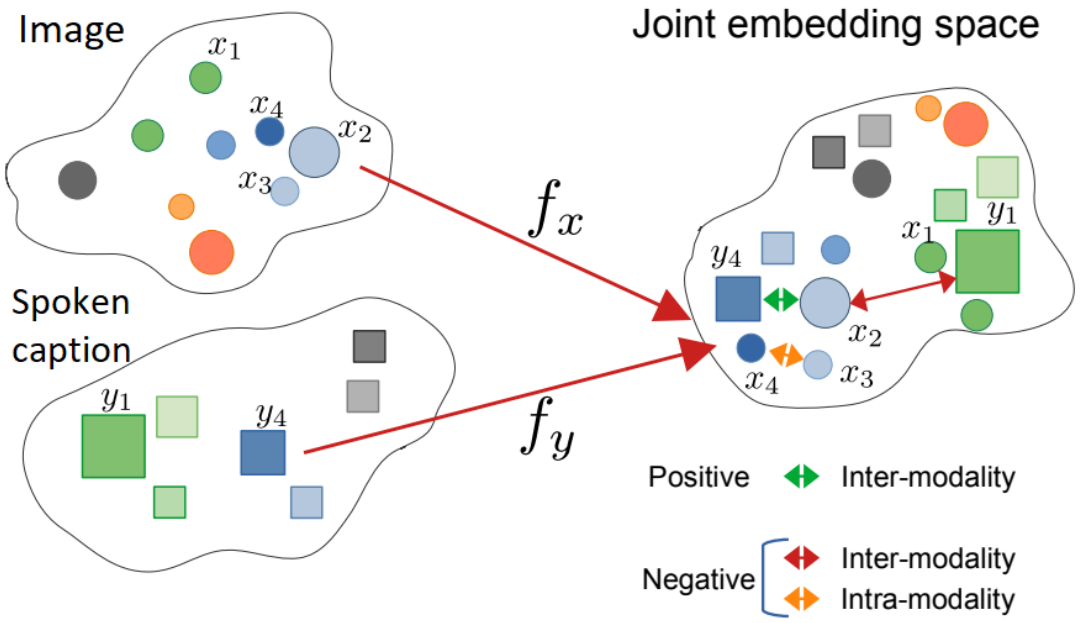

Figure 2.6. *Cross-modal contrastive learning. Semantically similar images and spoken captions share the same color scheme. Adapted from source [29].*

One common contrastive loss method is a “triplet” loss that is also widely used in VGS models. In VGS models, triplet loss tries to assign a higher similarity score to a semantically related image and spoken caption pair than to an unrelated pair [30]. A triplet set is made by taking one matching image-speech pair as “an anchor pair”. In addition, for the anchor image a random (unmatched) spoken caption is taken and for the anchor speech a random (unmatched) image is selected. This results in a total of one ground truth pair and two “impostor” pairs. The triplet loss function can be defined as

$$L(\theta) = \sum_{j=1}^{b} \max\left(0, S_j^c - S_j^p + M\right) + \max\left(0, S_j^i - S_j^p + M\right), \tag{2.8}$$

where S_j^p is the similarity score of the *j*th ground truth pair, S_j^c the similarity score between the anchor image and random caption, S_j^i the similarity score between the anchor speech with a random image, and *M* is the margin of the loss.

2.7 Evaluation metrics

VGS model performance is evaluated through two different retrieval tasks: speech-to-image retrieval, and image-to-speech retrieval. In the image-to-speech retrieval task, an image is given to the model as a query and the task of the model is to retrieve the correct spoken caption describing the query image. The speech-to-image retrieval task works similarly but vice versa, where the model is given a spoken caption as a query and its task is to retrieve the correct image.

The recall@k metric is used to measure the performance of the retrieval task in the VGS models. The *k* in recall@k stands for *k* nearest matches. I.e., if in the “image-to-speech” search the given query image and its matching spoken caption are in the top *k* nearest retrieved samples, the retrieval task is counted as successful. The distances between pairs are calculated from the embedding vectors and by using a similarity score such as cosine similarity. In the VGS models recall@1, recall@5, and recall@10 are widely used as the performance measurement metrics.

3. EXPERIMENTS AND RESULTS

This chapter introduces all the models that I designed, describes the experimental setup used, and discusses all the results that were yielded.

3.1 Datasets

The datasets I used for all the experiments were MSCOCO and SPEECH-COCO [7][8]. As mentioned in section 1.1, the SPEECH-COCO dataset synthesized speech captions for over 600k text captions for 123 287 different images in MSCOCO. Since my goal was to compare model performances, a smaller portion of the dataset was used. The training set included 40 000 images with 200 000 corresponding spoken captions and the validation set consisted of 10 000 images with 50 000 spoken utterances.

3.2 Models

I used the CNN0 model presented at [31] as the baseline model and tried to design some variations to that by modifying the speech encoder block. In the baseline model, the speech encoder is a convolutional block with five convolutional layers where except for the first convolutional layer, all convolutional layers are followed by a max-pooling layer (Figure Figure 3.1 left). The baseline model has increasing temporal receptive fields (RFs) meaning that each convolutional layer sees a larger region of the input speech data than the previous convolutional layer does. This audio CNN model is originally adopted from the work done in [11].

The first model variation I designed and tested was “Model 1” (Figure Figure 3.1 right). The architecture of Model 1 followed a similar layer structure as the baseline model with one additional convolutional layer and one additional max-pooling layer at the end. In total, Model 1 consisted of six convolutional layers and four max-pooling layers with increasing RFs. The main hyperparameter changed and tested with Model 1 was the kernel size which was reduced in convolutional layers. The last convolutional layer in Model 1 had 1024 filter channels which was double the number of filter channels in the last convolutional layer in the baseline model. Applying smaller kernel sizes in Model 1 resulted in reduced temporal receptive fields when comparing it to the baseline model. The motivation behind Model 1 was to see how the lower RFs affected the model’s performance.

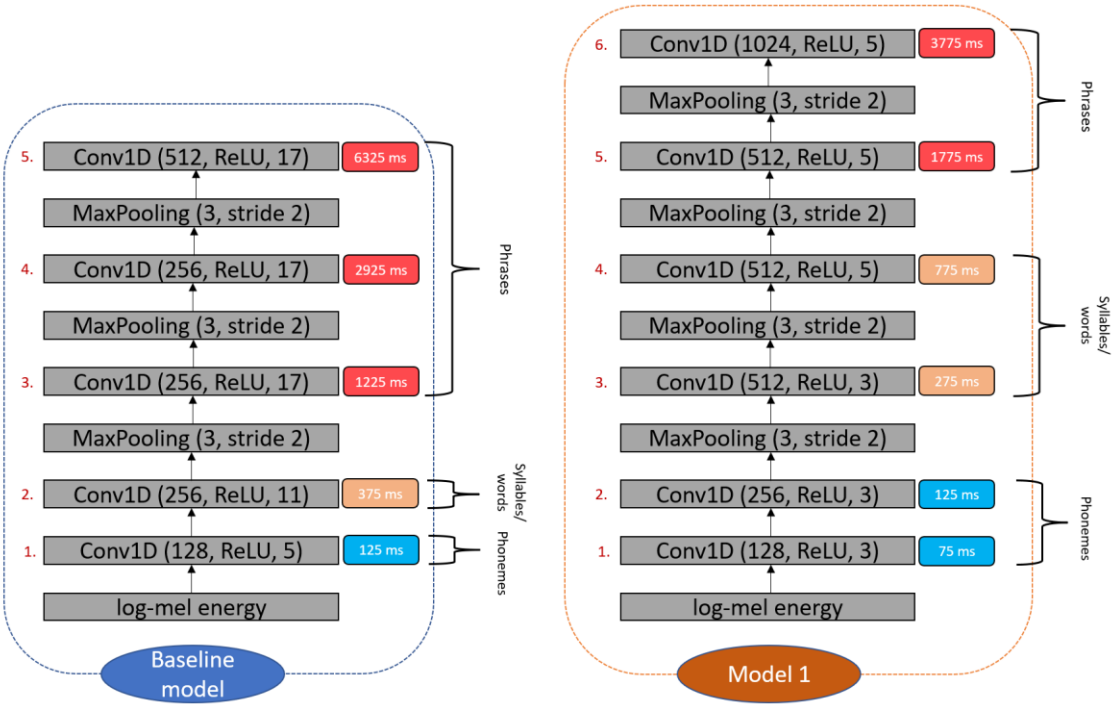

Figure 3.1. *Baseline and Model 1 convolutional audio encoder architecture and receptive fields in each layer.*

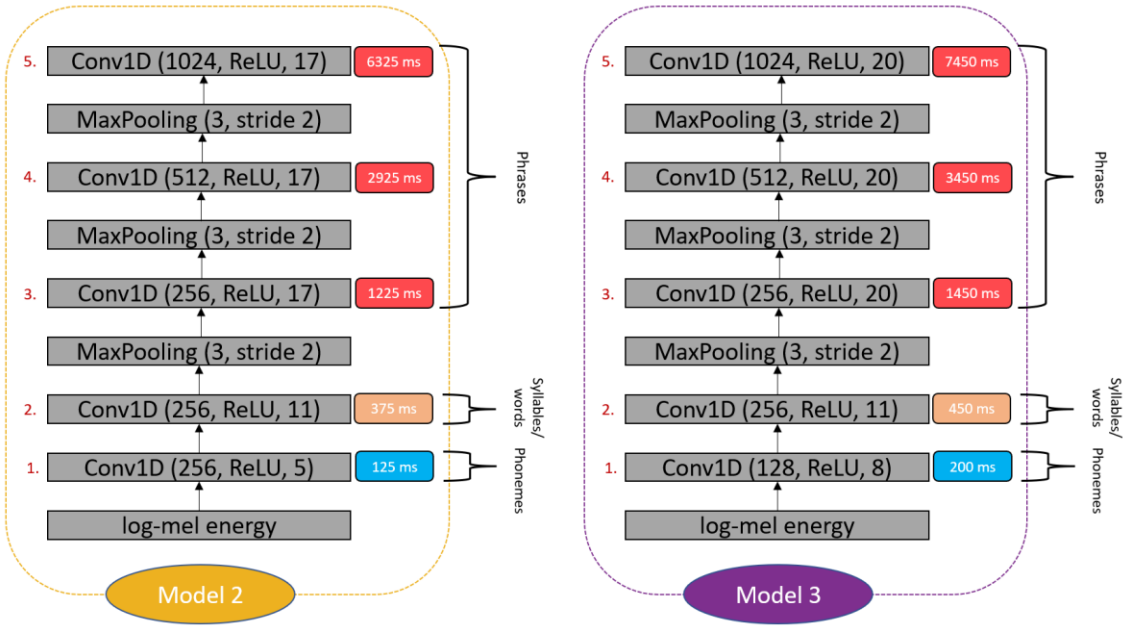

Figure 3.2. *Model 2 and Model 3 convolutional audio encoder architecture and receptive fields in each layer.*

The second model variation, “Model 2” (Figure Figure 3.2 left), has the same structure and RFs as the baseline model. The difference between Model 2 and the baseline model was the number of filter channels. I doubled the number of filter channels in the baseline model’s convolutional layers one, four, and five which resulted the Model 2. The idea

behind Model 2 was to see if the increased number of filter channels had a positive effect on the model's performance.

The third model variation that I designed and experimented with was "Model 3" (Figure Figure 3.2 right). Model 3 used the same architecture from the baseline model of five convolutional layers where except for the first convolutional layer, each convolutional layer is followed by a max-pooling layer. Model 3 had increased kernel sizes compared to the baseline model in all convolutional layers except the second convolutional layer. This resulted in higher temporal RFs which was the incentive in Model 3.

3.3 Experimental setup

The input features used in all the experiments were the same. Image features relied solely on a pre-trained VGG 16-layer network [32]. The models used in this work used the output of the last convolutional layer of VGG16 to represent image features. The final feature dimension for images was 14*14 and for speech 40-dimensional log-mel energies with 25-ms windows, 10-ms window hop-size, and 25-ms frame size.

All models used cosine similarity as a similarity score to map image and speech branches together and all models were trained using triplet loss with a margin $M = 0.1$. As an optimizer, Adam was used with a learning rate of 1e-4. The performance metric I measured and compared among all models was recall@10. All tests were run for 50 epochs and done on a computer with NVIDIA GTX 1080Ti graphics processing unit (GPU), 16GB of DDR4 random access memory (RAM), and running on Windows 10 operating system.

3.4 Results

Table Table 3.1 compiles the parameter count and retrieval task results for all compared models.

Table 3.1. *Parameter count and recall@10 scores for speech-to-image and image-to-speech retrieval tasks for all the compared models at 50 epochs.*

Model	Parameter count	Speech-to-image	Image-to-speech
Baseline	11,082,368	0.358	0.349
Model 1	8,915,840	0.281	0.287
Model 2	16,190,208	0.355	0.368
Model 3	17,981,568	0.393	0.386

All tested models were validated after each epoch using the validation data. Figure Figure 3.3 illustrates the training and validation loss curves for all tested model variations. Moreover, recall@10 calculated after each epoch for both speech-to-image and image-to-speech retrieval tasks are shown in figure Figure 3.4.

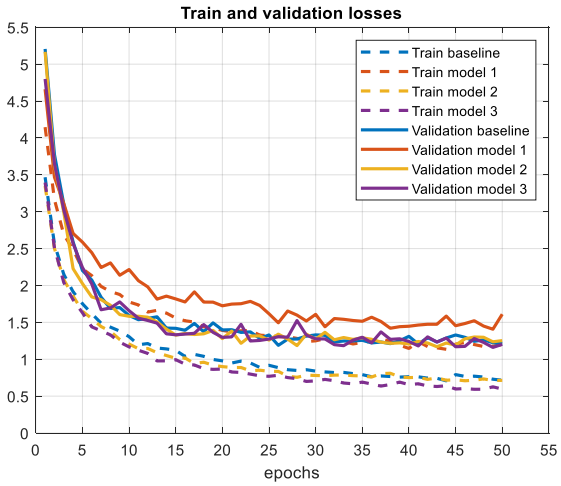

Figure 3.3. *Training and validation losses for all tested models.*

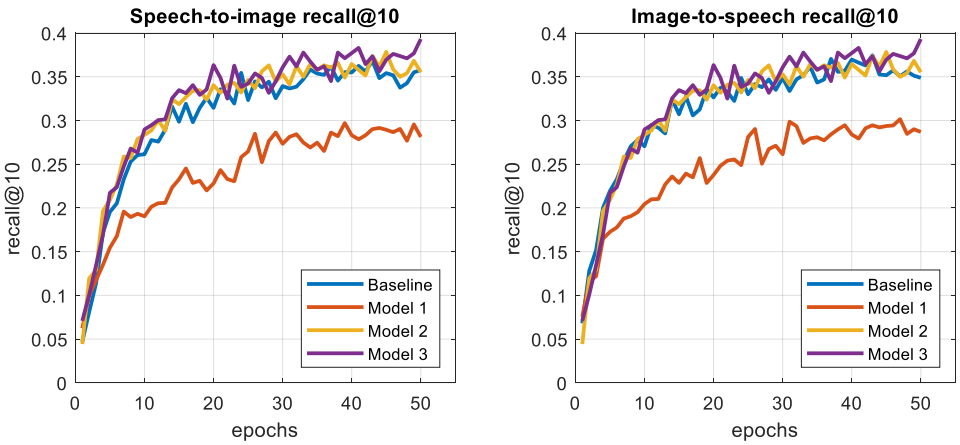

Figure 3.4. *Speech-to-image and image-to-speech recall@10 results for all tested models.*

The results show that all models learned to solve cross-modal retrieval tasks. Since the RFs of the convolutional layers in Model 1 were considerably lower than in the baseline model, the results show that reducing the temporal receptive fields negatively affect the model performance. Model 2 had identical RFs that the baseline model had. Results show that just increasing the number of filter channels and therefore model parameter count does not have a noticeable difference in model performance. Combining the higher number of filter channels in deeper convolutional layers and higher RFs than the baseline model, Model 3 had the best-performing results. It shows that some speech encoder optimization can be done to increase the model performance.

4. CONCLUSIONS

In this thesis, I studied the visually grounded speech models as neural methods for learning correspondences between image and speech data. More specifically, I focused to implement and design new model variations to the previous literature by modifying the speech encoder block in a CNN-based VGS model. However, due to the lack of proper computational resources, all experiments are applied to a small portion of data applied commonly in literature.

Overall, the results gathered in this thesis show that modifying the hyperparameters of the convolutional speech encoder block had clear effects on the model performance. Model 3 had the best performance in this study mainly due to the high temporal receptive fields of the convolutional layers.

Further research could be done either by testing and running baseline, Model 2, and Model 3 further using the whole SPEECH-COCO dataset, or by further developing the best performing model (I.e., Model 3) to upgrade its performance to an even higher level. Further studies could also be done to experiment with and develop more recent RNN and Transformer-based audio encoders discussed in Chapter 1.

REFERENCES

[1] Chrupała, G., 2022. Visually grounded models of spoken language: A survey of datasets, architectures and evaluation techniques. Journal of Artificial Intelligence Research, 73, pp.673-707.

[2] Dupoux, E., 2018. Cognitive science in the era of artificial intelligence: A roadmap for reverse-engineering the infant language-learner. Cognition, 173, pp.43-59.

[3] Merkx, D., Scholten, S., Frank, S.L., Ernestus, M. and Scharenborg, O., 2022. Modelling word learning and recognition using visually grounded speech. Cognitive Computation. 1-17. 10.1007/s12559-022-10059-7.

[4] Roy, D.K. and Pentland, A.P., 2002. Learning words from sights and sounds: A computational model. Cognitive science, 26(1), pp.113-146.

[5] Yu, C., Ballard, D.H. and Aslin, R.N., 2005. The role of embodied intention in early lexical acquisition. Cognitive science, 29(6), pp.961-1005.

[6] Rashtchian, C., Young, P., Hodosh, M. and Hockenmaier, J., 2010, June. Collecting image annotations using amazon's mechanical turk. In Proceedings of the NAACL HLT 2010 workshop on creating speech and language data with Amazon's Mechanical Turk (pp. 139-147).

[7] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C.L., 2014. Microsoft coco: Common objects in context. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13 (pp. 740-755). Springer International Publishing.

[8] Havard, W., Besacier, L. and Rosec, O., 2017. SPEECH-COCO: 600k Visually Grounded Spoken Captions Aligned to MSCOCO Data Set. 42-46. 10.21437/GLU.2017-9.

[9] Hsu, W.N., Harwath, D., Miller, T., Song, C. and Glass, J., 2021. Text-free image-to-speech synthesis using learned segmental units. 5284-5300. 10.18653/v1/2021.acl-long.411.

[10] Harwath, D. and Glass, J., 2015, December. Deep multimodal semantic embeddings for speech and images. In 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU) (pp. 237-244). IEEE.

[11] Harwath, D. and Glass, J.R., 2017. Learning word-like units from joint audio-visual analysis. 506-517. 10.18653/v1/P17-1047.

[12] Chrupała, G., Gelderloos, L. and Alishahi, A., 2017. Representations of language in a model of visually grounded speech signal. 613-622. 10.18653/v1/P17-1057.

[13] Chrupała, G., Higy, B. and Alishahi, A., 2020. Analyzing analytical methods: The case of phonology in neural models of spoken language. 4146-4156. 10.18653/v1/2020.acl-main.381.

[14] Peng, P. and Harwath, D., 2022, May. Fast-slow transformer for visually grounding speech. In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 7727-7731). IEEE.

[15] Mahesh, B., 2020. Machine learning algorithms-a review. International Journal of Science and Research (IJSR). [Internet], 9, pp.381-386.

[16] Sosnovshchenko, O. and Baiev, O., 2018. Machine Learning with Swift: Artificial Intelligence for IOS. Packt Publishing Ltd.

[17] Taylor, P., 2022. Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2020, with forecasts from 2021 to 2025, Statista. URL: https://www.statista.com/statistics/871513/worldwide-data-created/

[18] Russell, S.J., 2010. Artificial intelligence a modern approach. Pearson Education, Inc..

[19] Ray, S., 2019, February. A quick review of machine learning algorithms. In 2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon) (pp. 35-39). IEEE.

[20] Zhang, A., Lipton, Z.C., Li, M. and Smola, A.J., 2021. Dive into deep learning.

[21] Dubey, S.R., Singh, S.K. and Chaudhuri, B.B., 2022. Activation functions in deep learning: A comprehensive survey and benchmark. Neurocomputing.

[22] Sazli, M.H., 2006. A brief review of feed-forward neural networks. Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering, 50(01).

[23] Bengio, Y., Goodfellow, I. and Courville, A., 2017. Deep learning (Vol. 1). Cambridge, MA, USA: MIT press.

[24] Maier, A., Syben, C., Lasser, T. and Riess, C., 2019. A gentle introduction to deep learning in medical image processing, Figure 6. Zeitschrift für Medizinische Physik, 29(2), pp.86-101.

[25] Khorrami, K. and Räsänen, O., 2021. Can phones, syllables, and words emerge as side-products of cross-situational audiovisual learning? - A computational investigation. 10.34842/w3vw-s845.

[26] Bäckström, T., 2019. Cepstrum and MFCC, Aalto University Wiki. URL: https://wiki.aalto.fi/display/ITSP/Cepstrum+and+MFCC

[27] Wyse, L., 2017. Audio spectrogram representations for processing with convolutional neural networks.

[28] Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C. and Krishnan, D., 2020. Supervised contrastive learning. Advances in neural information processing systems, 33, pp.18661-18673.

[29] Zolfaghari, M., Zhu, Y., Gehler, P. and Brox, T., 2021. Crossclr: Cross-modal contrastive learning for multi-modal video representations, Figure 2. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 1450-1459).

[30] Harwath, D., Torralba, A. and Glass, J., 2016. Unsupervised learning of spoken language with visual context. Advances in Neural Information Processing Systems, 29.

[31] Khorrami, K. and Räsänen, O., 2021. Evaluation of audio-visual alignments in visually grounded speech models. 10.21437/Interspeech.2021-496.

[32] Simonyan, K. and Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. 3rd International Conference on Learning Representations (ICLR 2015), Computational and Biological Learning Society, 2015, pp. 1–14.