Multimodal Learning: Examples in Gesture and Audio-Visual Speech Recognition

Hsieh Yu-Guan

August 9, 2017

Abstract

1 Introduction

2 Related Work

3 Presentation of Basic Network Architectures

3.1 Conolutional Neural Networks

Convolutional Neural Networks (CNNs) are an early family of deep learning architectures inspired from the human vision system [15]. Generally we have convolutional layers alternating with pooling (subsamping) layers, but fully connected layers can also be introduced. CNNs have been shown to achieve state-of-the-art performance in image processing tasks such as image classification [14] and object detection [16]. However they can be equally applied in other fields like speech recogntion [6].

3.2 Auto-encoder

Auto-encoders are networks that are trained to minimize the reconstruction error by back-propagating it from the output layer to hidden layers. In the simplest model with one hidden layer, an auto-encoder takes an input $\mathbf{x} \in \mathbb{R}^d$ and maps it to the latent representation $\mathbf{h} \in \mathbb{R}^{d'}$ given by $\mathbf{h} = \sigma(W\mathbf{x} + \mathbf{b})$ where W is a weight matrix, $\mathbf{b}$ is a bias vector and σ is an activation function. Then the network tries to reconstruct the input by a reverse mapping $\mathbf{x}' = \sigma(W'\mathbf{h} + \mathbf{b}')$.

To prevent the auto-encoder from learning the identity function as a trivial solution, several regularization techniques have been proposed. The bottleneck approach forces dimensionality reduction by having fewer neurons in hidden layers than in the input layer. For example, in the above case, we must have $d' < d$. Sparse auto-encoders impose sparsity on hidden units [17]. Denoising auto-encoders, which play an important role in my internship, try to recontruct the clean input from its corrupted version [3, 39]. Binomial noise (switching pixels on or off) adding to input or hidden layers are used in my case.

Intuitively, auto-encoders are useful for data reconstruction. Nevertheless, the true interest lies in fact in its capcity to learn a representation (encoding) for a set of data in a purely unsupervised fashion [40]. Recently, auto-encoders are also more and more often used as a generative model [4].

4 Datasets and Preprocessing

Many datasets were explored during my internship. The three main datasets being used are given in details below. Two of them are for gesture recognition: Creative Senz3D [20, 21] and ASL Finger Spelling [32], and one is for AVSR: AVLetters [19].

4.1 Creative Senz3D

The dataset contains gestures perfomed by 4 different people, each performing 11 different static gestures repeated 30 times each, for a total of 1320 samples. For each sample, color, depth and confidence frames are available. I only used the color and depth frames of this dataset. The original size of each image is 480 × 640 and they're resized to 299 × 299 pixels before being fed to the network. No other preprocessing are done. For both color and depth images I use the three color channels (even though a priori only one channel is needed for depth maps).

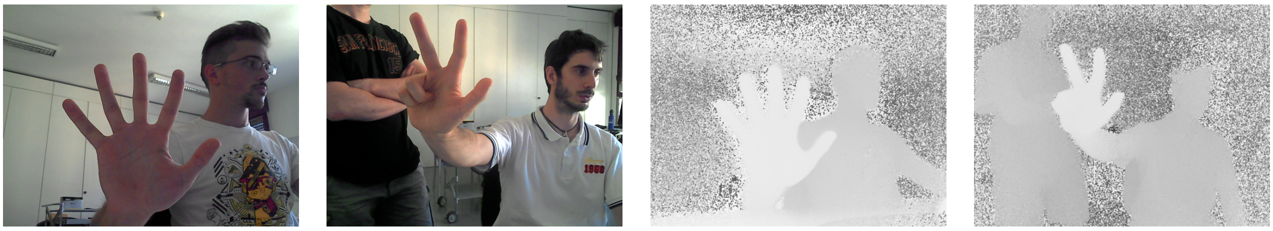

Figure 1: Example images in the Creative Senz3D dataset.
Left Two) Color images.
Right Two) Corresponding depth images.
All of the images are of size 480 × 640 and contain the the entire upper body of the subject.

4.2 ASL Finger Spelling

The dataset is composed of more than 60000 images in each modality (RGB and depth images are provided). Five subjects are asked to perform the 24 static signs in the American Sign Language (ASL) alphabet (excluding j and z which involve motion) a certain number of times, captured with similar lighting and background.

Images of this dataset are of variable sizes. The data preprocessing includes resizing each image to 83 × 83 pixels, converting to grayscale and adjusting contrast of depth maps. Only very late in my internship I added the Z-normalization (normalize to zero mean and unit of variance) as a preprocessing step and the only result that was largely changed is presented in 6.2.

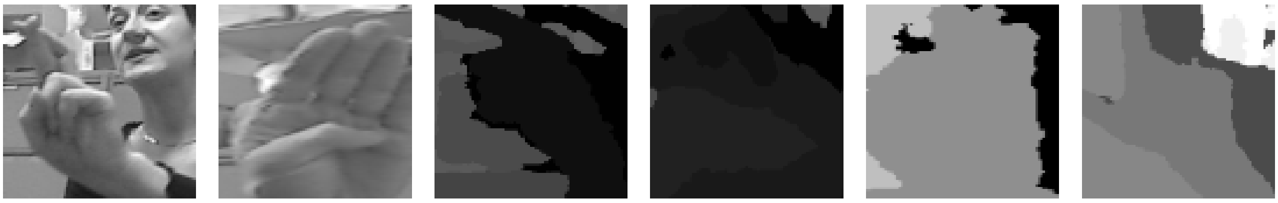

Figure 2: Example images in the ASL Finger Spelling dataset (after preprocessing).
Left Two) Grayscale intensity images.
Middle Two) Depth maps after adjusting contrast.
Right Two) Depth maps after Z-normalization.
Images of this dataset have variable sizes, and they're all resized to 83 × 83 before being fed to the network. Generally only the hand region is contained in image.

4.3 AVLetters

The dataset comprises video and audio recordings of 10 speakers uttering the letters A to Z, three times each. We count therefore 780 samples in total. For video data, image sequences of pre-extracted lip regions are provided. Each single image if of size 60 × 80. For audio data, only the mel-frequency cepstrum coefficients (MFCCs) are given, and each audio frame is represented by 26 mfccs. The lack of raw audio data is a strong constraint on what we're able to do on this dataset.

Since all utterances don't have the same time duration, I used fourier resamping to force every video input to be of length 12 and every audio input to be of length 24. Video frames are Z-normalized. Several data augmentation techniques are also considered, including random brightness adjusting, random contrast adjusting and random cropping (but at least 80% of the original image is kept).

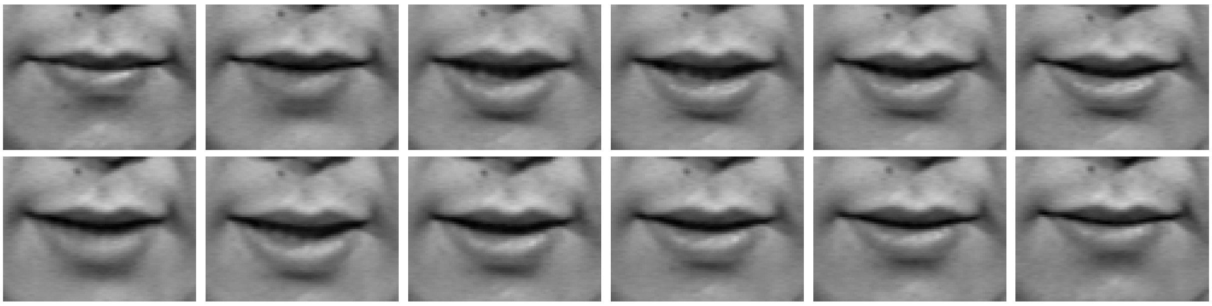

Figure 3: Example visual input for the AVletters dataset (left to right, top to bottom). Pre-extracted lip regions of 60×80 pixels are provided. Each image sequence is resampled to be of length twelve in order to give an input of fixed size to the network.

5 Experimental Setup

To train a classifier I employed the cross entropy cost function and to train an auto-encoder the L2 distance between the input and output vector was used as the loss. For the sake of preventing overfitting, L2 regularization [3] was applied to all the weights of network with a regularization coefficient of 0.0004. The Adam algorithm [13] was then introduced for minimizing the loss function. An exponential decay was further used for the stepsize α of this algorithm with an initial stepsize α_0 varying from 0.01 to 0.0001 depending on experience. The decaying rate γ was generally close to 0.8 and the decay takes place every 100 training steps.

Inputs of the network are normally fed as mini-batches of size 24 (smaller and bigger batch sizes were also experimented). Batch normalization [10] were introduced after every convolutional and transposed convoluitonal layer. Therefore, the real operations used to compute neural activations are more complicated then what are described above. These settings aren't necessarily optimal (use a bigger weight regularization coefficient or maybe one had better use a constant learning rate with Adam optimizer) but I didn't have time to do more tests on it.

Here are some more details of the network architectures: ReLu (Rectified Linear Unit) activation were added to all the hidden layers [14] while no activation function was used for the output layer. For classification model dropout [34] was always applied to the second to last layer during training. The output of the classification layer was mapped to the probabilities that one data example belongs to each class by the softmax function.

6 Experiences and Results: Unimodal Cases

6.1 Classification

With every new dataset, I began with training a classifier on it in a totoally supervised manner. This gave me an insight into its data quality, the preprocessing effectiveness and ensured that further experiments could be conducted. CNN is then one of the most suitable architecture for this purpose.

6.1.1 Creative Senz3d

No satisfying results were acquired. It may be due to to a lack of data quantity, variety, and the fact that the head is also contained in the image increases significantly the classification difficulty.

Subject Dependent. In a subject dependant setting, images are separated randomly into training set (3/4) and test set (1/4). Therefore, during the testing phase, the classifier doesn't need to deal with data from an individual that it has never seen before. In this case, for RGB images, all of the classifiers are able to have a classification accuracy that is closed to 100%. This holds true even for a perceptron. On the other hand, for depth images, the classification accuracy is between 60% and 70% using a perceptron and near 90% for other CNN architectures that were tested.

Subject Independant. On the contrary, the classifier faces individuals never seen before during testing in a subject independant setting. In my case, the training set consists of images coming from the first three individuals while the test set contains images of the final subject. With the various architectures (including single-layer perceptron and CNNs varying from three to ten hidden layers) that I implemented, none of them is able to generalize the learned model to the new individual. The pre-trained InceptionV4 architecture achieves a prediction accuracy of 30% for color images and 20% for depth images (better than chance).

6.1.2 ASL Finger Spelling

The large number of data contained in this dataset and the relatively simple image content (single hand instead of the entire upper body) makes the classification task much easier. By using the CNN architecture shown in Figure 4, we can achieve a classification accuracy of respectively 80% and 70% for intensity and depth images (Table 1) in an subject-independant setting (four subjects for training and one subject for testing). We may not need that many layers in the CNN architecture, but further tests were not carried out since it's not the essential point of my internship.

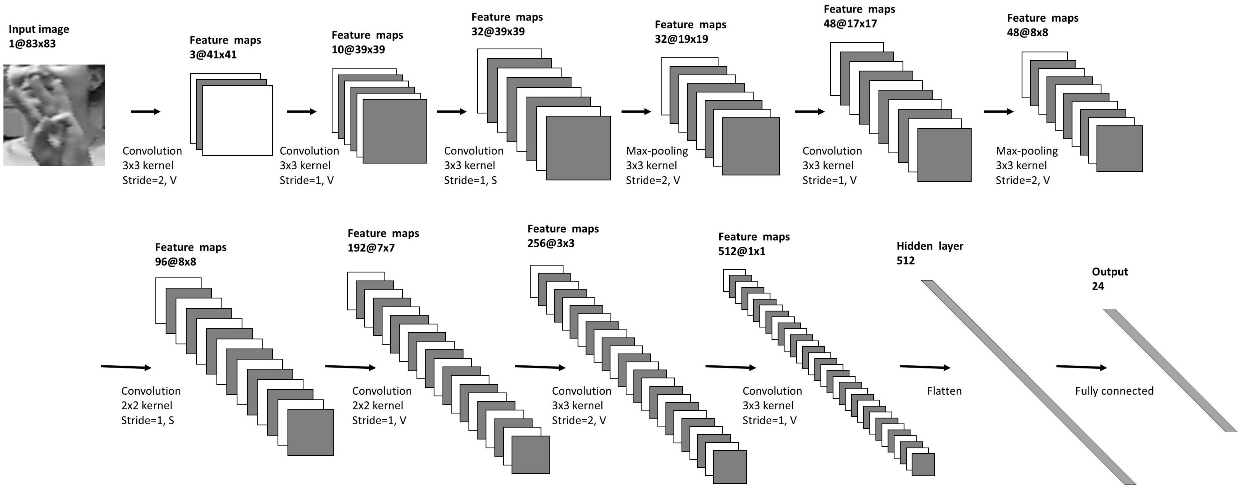

Figure 4: **CNN architecture used for the Finger Spelling dataset.**
The input of the nework is a one-channel image of size 83 × 83. It contains ten hidden layers. S stands for 'SAME' padding and V stands for 'VALID' padding (see text).

6.1.3 AVletters

One can refer to Figure 9 for the main CNN architectures that are used in this dataset. Notice that 3d CNNs are employed to deal with video inputs. Considering the small number of available data, a speaker-dependant setting was used, but it didn't save me from the problem of overfitting. The classification accuracy is of 100% for training data but only of 60% or 55% for testing data depending on the input modality (audio then video).

Curiously, for audio data, I get exactly the same classification performance regardless of the used architecture (perceptron or CNNs) or the fact that if deltas and delta-deltas are also given in input. This is not the case when I test with another audio dataset (not mentioned in the Datasets and Preprocessing section beacause it wasn't used for main experiences). For video input, the use of data augmentation techniques only decrease the learning speed for the training part but doesn't improve the performace for testing.

6.2 Convolutional auto-encoder

Several different CAE architectures were tested in my internship. Here I present the one with five hidden layers, therefore it contains three convolutional layers and three transposed convolutional layers as shown in Figure 5. The end-to-end training instead of a greedy layer-wise approach was used.

The proposed architecture was then trained on the two gesture recognition datasets. First of all, I'm interested in the denoising capcity of the auto-encoder. An example is shown in Figure 6. The auto-encoder is effectively able to reconstruct the clean image in a way.

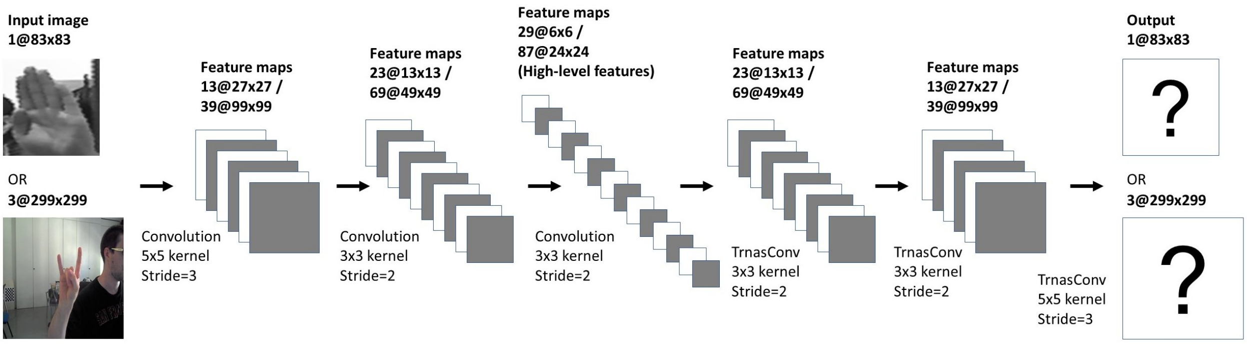

Figure 5: Convolutional auto-encoder architecture with three convolutional layers and three tranposed convolutional layer.
Activation values of the middle layer are taken as high-level features of the input image. Inputs of the network can be of different sizes. We only use valid paddings here.

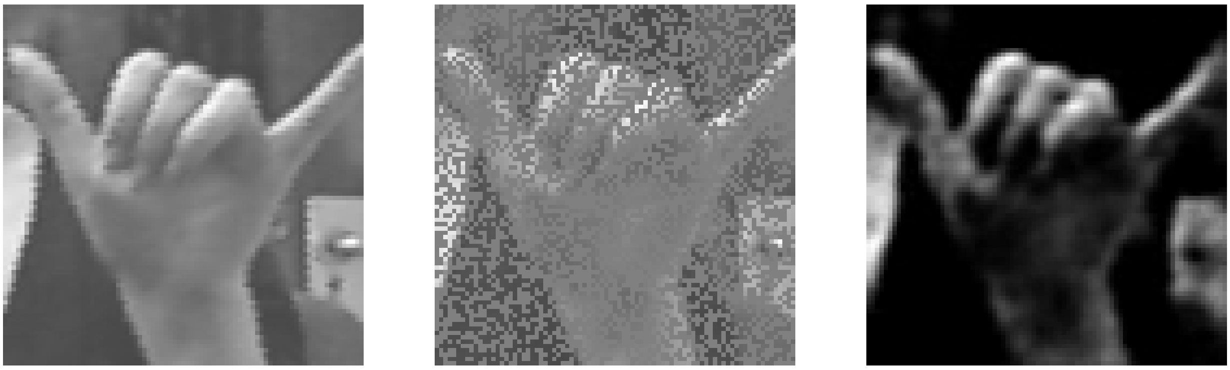

Figure 6: Image restoration using convolutional auto-encoder.
Left) Clean Image.
Middle) Noisy image [input].
Right) Restored image [output].

7 Experiences and Results: Multimodal Cases

7.1 Learning shared representation

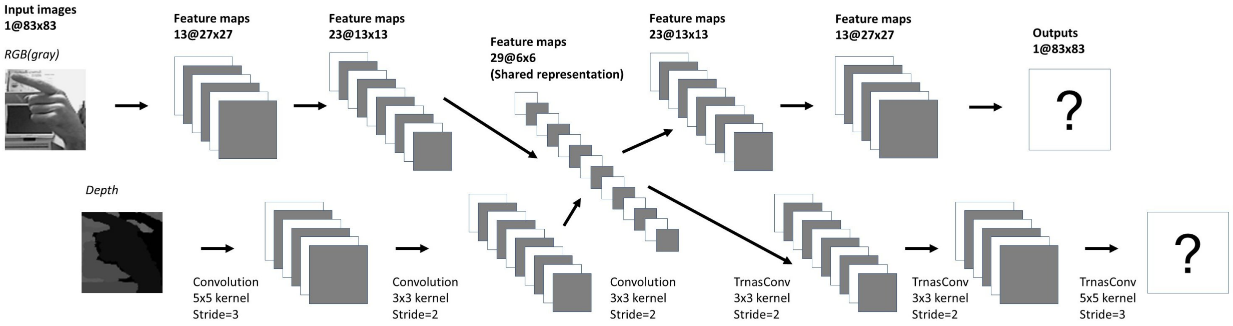

Figure 7: The bimodal convolutional auto-encoder model that is used to learn shared multimodal representation.
We simply take the CAE architecture that is introduced earlier (Figure 5) for each modaliy but force them to have a shared middle layer by adding the corresponding activation values. We then try to reconstruct the two images separately through two disjoint paths.

		Raw	CAE features	Shared	CAE architecture	CNN
Intensity	train	69.47 %	78.39 %	81.07 %	92.34 %	99.69 %
	test	32.64 %	50.72 %	50.92 %	67.23 %	79.73 %
Depth	train	48.17 %	38.65 %	73.07 %	90.4 %	97.24 %
	test	30.95 %	26.39 %	43.50 %	58.69 %	66.84 %
Depth (Z-n)	train	63.64 %	72.37 %	72.01 %	%	97.24 %
	test	29.93 %	40.03 %	40.27 %	%	64.46 %

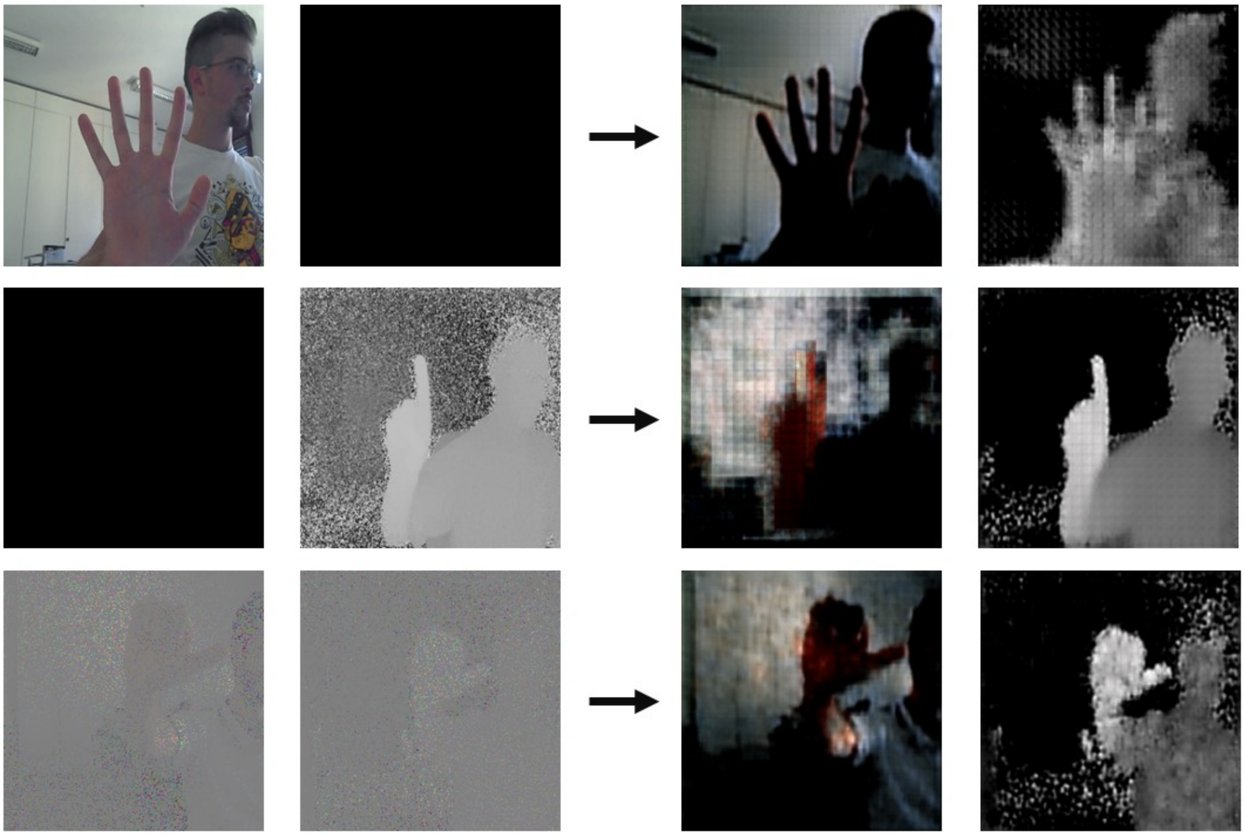

Figure 8: **Restore color and depth images from incomplete input information.**
Top) Only the color image is given.
Middle) Only the depth image is given.
Botttom) Both modalities are given but with little information (10% of pixels).

7.2 Transfer learning

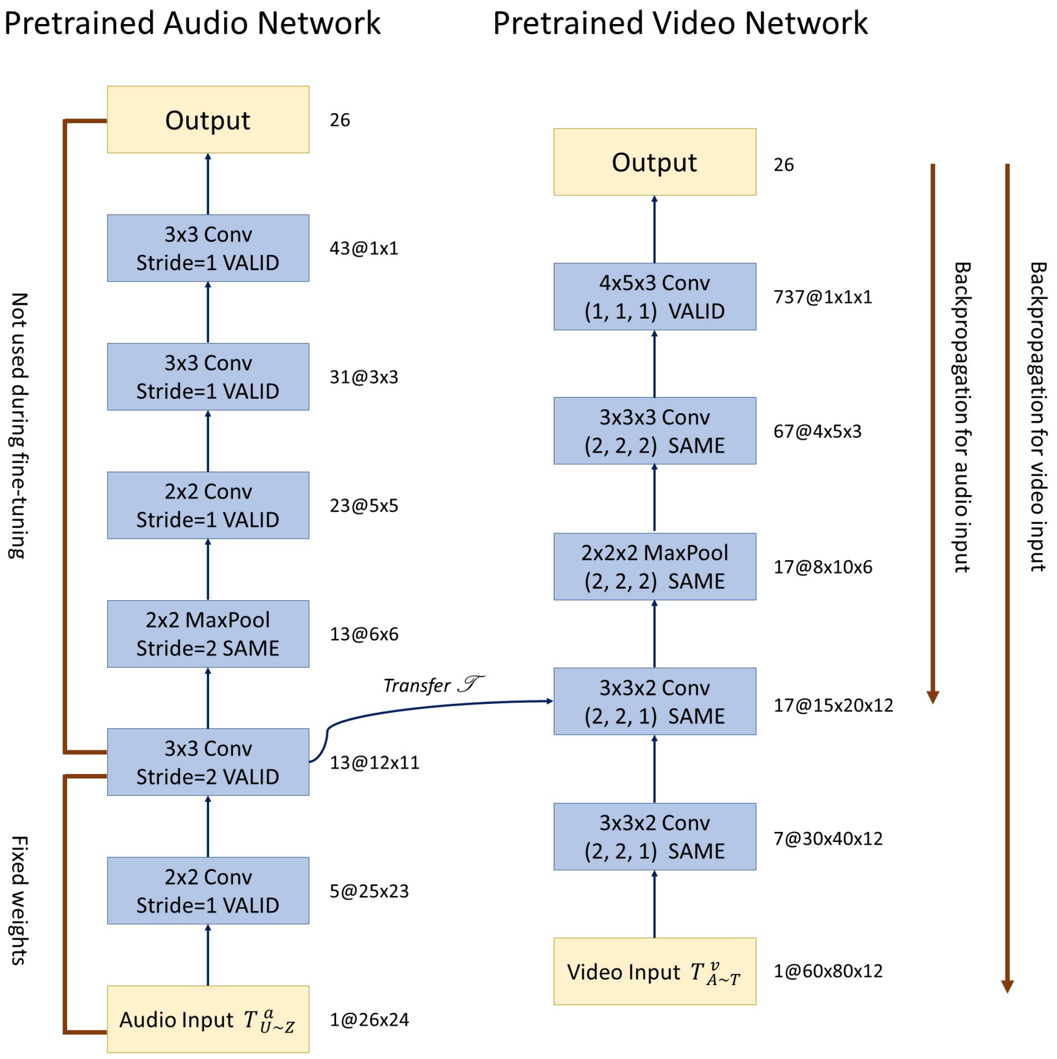

Figure 9: Illustration of the transfer learning approach applied in audio and lip-reading speech recognition tasks.
We first learn two separated model for audio and visual inputs (in my cases two CNNs) and try to fine-tune the video network with transferred audio data.

References

[1] 17. M. Asadi-Aghbolaghi, A. Clapés, M. Bellantonio, H. J. Escalante, V. Ponce-López, X. Baró, I. Guyon, S. Kasaei, and S. Escalera. A Survey on Deep Learning Based Approaches for Action and Gesture Recognition in Image Sequences. In *Automatic Face & Gesture Recognition (FG 2017) 2017 12th IEEE International Conference*, 2017.

[2] T. Baltrusaitis, C. Ahuja and L-P. Morency. Multimodal Machine Learning: A Survey and Taxonomy. In *arXiv preprint arXiv: 1705.09406*, 2017.

[3] Y. Bengio. Practical recommondations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pages 437–478, 2012.

[4] Y. Bengio, L. Yao, G. Alain and P. Vincent. Generalized denoising auto-encoders as generative models. In *Advances in Neural Information Processing Systems*, pages 899–907, 2013.

[5] C. Bregler and Y. Konig. "Eigenlips" for robust speech recognition. In *ICASSP*, 1994.

[6] L. Deng, G. Hinto and B. Kinsbury. New types of deep neural network learning for speech recognition and related applications: An overview. In *ICASSP*, 2013.

[7] A. Droniou, S. Ivaldi, and O. Sigaud. A deep unsupervised network for multimodal perception, representation and classification. In *Robotics and Autonomous System*, 2014.

[8] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013.

[9] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition. In *IEEE Signal Processing Mag* Vol. 29, 2012.

[10] S. Ioffe and C. Szegedy. Batch normalization, accelerating deep network training by reducing internal covaraiate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[11] S. Ji, W. Xu, M Tang and K. Yu. 3D Convolutional Neural Networks for Human Action Recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, 2013.

[12] A. K. Katsaggelos, S. Bahaadini and R. Molina. Audiovisual fusion: Challenges and new approaches. In *Proceedings of the IEEE*, vol. 103, 2015.

[13] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *arXiv preprint arXiv:1412.6980*, 2014.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, 2012.

[15] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, vol. 86, pages 2278–2324, 1998.

[16] Y. LeCun, K. Kavukcuoglu and C. Farabet. Convolutional networks and apllications in vision. In *Circuits and Systmes (ISCAS), Proceedings of 2010 IEEE International Symposium*, pages 253–256, 2010.

[17] A. Makhzani and B. Frey, K-Sparse autoencoders. In *International Conference on Learning Representations (ICLR)*, 2014.

[18] J. Masci, U. Meier, D. C. san, and J. Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *Artificial Neural Networks and Machine Learning–ICANN*, 2011.

[19] I. Matthews, T. F. Cootes, J. A. Bangham and S. Cox. Extraction of visual features for lipreading. In *PAMI*, 2002.

[20] A. Memo, L. Minto and P. Zanuttigh. Exploiting Silhouette Descriptors and Synthetic Data for Hand Gesture Recognition. In *STAG: Smart Tools & Apps for Graphics*, 2015.

[21] A. Memo and P. Zanuttigh. Head-mounted gesture controlled interface for human-computer interaction. In *Multimedia Tools and Applications*, 2017.

[22] S. Mitra and T. Acharya. Gesture recognition: A survey. In *IEEE Systems, Man, and Cybernetics*, 37:311–324, 2007.

[23] P. Molchanov, S. Gupta, K. Kim, and J. Kautz. Hand gesture recognition with 3D convolutional neural networks. In *CVPRW*, 2015.

[24] S. Moon, S. Kim and H. Wang. Multimodal transfer deep learning with applications in audio-visual recognition. In *NIPS MMML Workshop*, 2015.

[25] J. Nagi, F. Ducatelle, G. Di Caro, D. Ciresan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, and L. Gambardella. Max-pooling convolutional neural networks for vision-based hand gesture recognition. In *Proceedings of the IEEE International Conference on Signal and Image Processing Applications*, 2011.

[26] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout. Multiscale deep learning for gesture detection and localization. In *ECCVW*, 2014.

[27] J. Ngiam, A. Khosla, M. Kim, J. Nam, and A. Y. Ng. Multimodal deep learning. In: In *International Conference on Machine Learning*. 28. Bellevue, Washington, USA, 2011.

[28] K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno and T. Ogata. Audio-visual speech recognition using deep learning. In *Applied Intelligence, vol. 42, no. 4, pp. 722–737, 2014.*

[29] E. Petahan, B. Bischoff, D. Bodoff, and N. M. Brooke. An Improved Automatic Lipreading System to enhance Speech Recognition. In *ACM SIGCHI*, 1988.

[30] L. Pigou, S. Dieleman, P. J. Kindermans, and B. Schrauwen. Sign language recognition using convolutional neural networks. In *Workshop at the European Conference on Computer Vision*, pages 572–578, 2014.

[31] G. Potamianos, C. Neti, J. Luettin and I. Matthews. Audio-visual automatic speech recognition: An overview. In *Issues in Visual and Audio-Visual Speech Processing, MIT Press*, 2004.

[32] N. Pugeault, and R. Bowden. Spelling It Out: Real-Time ASL Fingerspelling Recognition. In *Proceedings of the 1st IEEE Workshop on Consumer Depth Cameras for Computer Vision*, 2011.

[33] R. Socher, M. Ganjoo, H. Sridhar, O.Bastani, C. D. Manning, and A. Y. Ng. Zero-shot learning through cross-modal transfer. In *International Conference on Learning Representations (ICLR)*, Scottsdale, Arizona, USA, 2013.

[34] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. In *Journal of Machine Learning Research*, vol. 15, 2014.

[35] T. Starner, A. Pentland, and J. Weaver. Real-time american sign language recognition using desk and wearable computer based video. In *PAMI*, 20(12):1371–1375, 1998.

[36] J. Sung, I. Lenz, and A. Saxena. Deep multimodal embedding: Manipulating novel objects with point-clouds, language and trajectories. In *Robotics and Automation (ICAR), 2017 IEEE International Conference*, pages 2794–2801, 2017.

[37] C. Szegedy, S. Ioffe, V, Vanhoucke. Inception-v4, inception-ResNet and the impact of residual connections on learning. In *AAAI*, pages 4278–4284, 2017.

[38] V. Turchenko, E. Chalmers and A. Luczak. A deep convolutional auto-encoder with pooling – unpooling layers in Caffe. In *arXiv preprint arXiv:1701.04949*, 2017.

[39] P. Vincent, H. Larochelle, Bengio and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine Learning (ICML)*, pages 1096–1103; 2008.

[40] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio and P.-A. Manzagol. Stacked denoising autoen-

codes: Learning useful representations in a deep network with a local denoising criterion? In *The Journal of Machine Learning Research*, vol. 11, pages 3371–3408, 2010.

[41] J. Weston, S. Bengio, and N. Usunier. Large scale image annotation: learning to rank with joint word-image embeddings. In *Machine Learning*, 81(1):21–35, 2010.

[42] Di Wu, Lionel Pigou, Pieter-Jan Kindermans, Nam Do-Hoang Le, Ling Shao, Joni Dambre, and Jean-Marc Odobez. Deep Dynamic Neural Networks for Multimodal Gesture Segmentation and Recognition. In *Pattern Analysis and Machine Intelligence IEEE Transactions*, vol. 38, 2016.

[43] B. P. Yuhas, M. H. Goldstein, and T. J. Sejnowski. Integration of acoustic and visual speech signals using neuralnetworks. In *IEEE Comm. Magazine*, pp. 65–71, 1989.