The effect of multimodality on the performance of a convolutional neural network.

Bachelor's Project Thesis

Danny Rogaar, dannyrogaar@gmail.com,
Supervisors: MSc M. Ameryan & Prof. dr. L.R.B. Schomaker

Abstract: A classical convolutional neural network is connected to two inputs of different modality (language, images, music, etc.) to observe the effect on accuracy. One modality used is iconographic images, representative of some concept. The second modality used is titles for the same concept, converted to images. The modalities are both processed using a convolutional neural network and a fully connected representation layer, with different parameters. Connecting the convolutional networks to a fully connected classification layer allows comparing the resulting bimodal classifier with control-condition networks. Initial results show that overfit representations disabled transfer learning which was required to connect the modalities. The perceived overfitting was counteracted using dropout, after which the bimodal network has significantly more accuracy (98.6%) than the control-conditions (93.6% on images and 93.0% on words) when given both inputs.

1 Introduction

The input space of neural networks is often bound to single type of information, due to the relative ease. However, humans tend to use all relevant types of input in perception, see e.g. Rosenblum (2008b) on arguments for this stance. Hence, researchers are combining multiple types of information in hopes of increased performance on their respective information processing model. Especially audiovisual speech processing has received much attention (see Schomaker, Nijtmans, Camurri, Morasso, Benoit, Guiard-Marigny, Gof, Robert-Ribes, Adjoudani, Defee, Munch, Hartung, and Blauert (1995) as well as Bailly, Perrier, and Vatikiotis-Bateson (2012)). Part of the reason for the focus on audiovisual processing is the strong relationship evident between speech and the speakers visible facial features, exemplified through the well-known McGurk effect observed by McGurk and MacDonald (1976). Given that humans are able to perform well in an environment with multiple types of information, this paper assumes a similar potential for computational models. Specifically, the paper investigates the potential of convolutional neural networks to effectively integrate two types of information.

For terminology used hereafter, modality indicates what type of information is conveyed, for example music and natural language. Distinctly, the medium indicates how the information is transferred, for example by image, sound or touch. Given the potential for increased performance in a multimodal environment, the research question here is what kind of effect multimodality has on neural network performance. The problem is specialised to classification on two different modalities. The modalities used will be (1) iconographic images and (2) the image titles, converted to images themselves (like written text). Thus, the two modalities use the same visual image medium. Since multimodal data contains more independent information than a unimodal set does, performance is expected to be higher on multimodal data. Moreover, in order to obtain such a better performance, then, features may incorporate information from both modalities and become richer in their representation. However, it is also possible that plain accuracy decreases whereas generalisation to unseen or noisy data improves, as seen in research on visual speech (Ngiam, Khosla, Kim, Nam, Lee, and NG (2011)). Followingly, the multimodal classifier is expected to have less accuracy than its unimodal variants, but show better generalisation to noisy or new data. The

hypothesis is tested by comparing the multimodal classifiers to unimodal comparison networks, as well as the original unimodal classifiers.

Since the used medium is images, classifiers in this research will process images using Convolutional Neural Networks (CNN), achieving the best performance on similar tasks, see e.g. the achievement by Krizhevsky, Sutskever, and Hinton (2012). A fully connected representation layer is added before the classification layer to stimulate multimodal connections and improve training, see also section 2.

1.1 Related work

As mentioned earlier, humans tend to use multiple modalities when dealing with perception of e.g. speech. The project by Schomaker et al. (1995) discusses and widely structures multimodal processing relevant to human-computer interactions. The technical report contains a more complete overview of multimodal processing than this document will. For an early historical overview on multimodal processing refer to Bernstein and Benoît (1996). A big question in (audiovisual) research has been to uncover how the multimodal information is combined, mainly distinguishing if integration takes place after or before feature extraction. See the research by Rosenblum (2008a) for a small argument on how visual and auditory information is concurrently processed by humans, and at what stage the modalities are combined. In line with late integration of multimodal data, this paper assumes the best model to process multiple inputs separately, before combining the resulting features. Likewise, Bengio (2004) apply hidden markov models on multimodal speech, processing the auditive and visual information in separate ways. The authors significantly improve accuracy with a higher margin as more noise is added to the speech. The research further highlights the advantage of training asynchronously between modalities, thereby showing the practical relevance of training before integration of modalities.

A foundation for multimodal deep learning is Ngiam et al. (2011) where greedy layer wise training (Hinton, Osindero, and Teh (2006)) is used for restricted Boltzmann machines on an autoencoding task (RBM, Smolensky (1986)). By stacking RBM's, deep belief networks by Hinton et al. (2006) are created for multimodal reconstruction. The authors find the deep autoencoders to perform best when multiple modalities are given during feature learning and training is performed in a unimodal fashion. When trained on both modalities, an audio-only RBM performs best. However, under noisy audio conditions, video features complement audio well to outperform both audio and video RBMs in accuracy. The paper shows that multimodal deep learning may increase performance of deep learning architectures, although the results are not consistent in every scenario. Also, the research inspired Huang and Kingsbury (2013) to use similar deep belief networks in an audiovisual recognition task. Moreover, in the described paper (Ngiam et al. (2011)), pretraining was used on the autoencoders for feature learning. However, since the paper by Glorot, Bordes, and Bengio (2011) showed that deep neural networks using Rectified Linear Units (ReLU) do not have to use pretraining in order to achieve top performance, the pretraining phase is often left out in modern CNN's. Currently many approaches use neural networks and often use them in combination with the successful CNN's for image processing. An example of research utilising neural networks is by Mroueh, Marcheret, and Goel (2015), who also use correlations between the modalities for further improvements.

Another inspiration for this research is the idea of using word embeddings in machine learning, corresponding to the earlier ideas of latent semantic analysis: Dumais (2004). The word embeddings are shown to have useful semantic properties, see Mikolov, Sutskever, Chen, Corrado, and Dean (2013b) who have also popularised the method with efficient implementations. The skip-gram (Mikolov, Chen, Corrado, and Dean. (2013a)) model used by Mikolov et al. maps sparse vectors representing words to dense semantic vector spaces. The dense vectors then hold the context for a given word. Research using word context include Frome, Corrado, Shlens, Bengio, Dean, Ranzato, and Mikolov (2013) and their more recent contribution Norouzi, Mikolov, Bengio, Singer, Shlens, Frome, Corrado, and Dean (2014). Both papers consider the same setting: classifying

images with the added word context modality. In Frome et al. (2013), the authors introduce the DeViSE model, where a pretrained convolutional neural model for images is then repurposed to predict dense word vectors. The nearest word vector(s) to the prediction then represents the predicted class(es), allowing prediction outside of the image dataset. Evaluating on top-k predictions, the model reached an accuracy equal to AlexNet (Krizhevsky et al. (2012)) when $k = 10$, increased accuracy for $k > 10$ and also shows increased generalisation. The ConSE architecture from Norouzi et al. (2014) uses a simpler method towards the same goal. Instead of predicting the dense vectors, the authors estimate the vector by taking the average of dense vectors weighed by the corresponding class probabilities, which are predicted by the visual classifier. Doing so requires no further training for the visual classifier. The multimodal architectures here efficiently leverage a different modality to increase generalisation, that is, the performance on data unknown to an already existing image classifier.

Another relevant concept to this research is disentangling factors of variation, basically learning useful features independent from the inputs or other features. Mainly, this project was purposed to find representations across concepts and, thus, disentangled with respect to concepts. The research also benefits from reducing the amount of information unrelated to the task, which is implied by disentangling factors of variation. In a paper by Larsen, Sønderby, Larochelle, and Winther (2016), the authors are able to show the possibility of learning semantically meaningful visual features as opposed to the more abstract features generally learned in CNN's. To the goal of feature learning, the pixel wise error in a traditional autoencoder is replaced with a feature-wise error. The feature-wise error is obtained from the discriminator of a Generative Adversarial Network (GAN, Goodfellow, Pouget-Abadie, Mirza, Xu, Warde-Farley, Ozair, Courville, and Bengio (2014)). By combining a variational auto-encoder (Kingma and Welling (2014)) with the GAN, they are able to separate the concepts of, for example, male/female or whether a person has glasses or not. As such, it became possible to draw e.g. glasses on an image of a person. This paper also hopes to find features that are better able to represent concepts, but by learning different modalities. Note that features in this paper do not aim to represent a visual feature explicitly, which may results in more abstract representations. However, if representation learning is successful in this project, the abstract representations are expected to be related to some independent and, thus, disentangled representations.

2 Methods

The problem in this paper is to design neural networks that are similar and will allow the comparison of a single modality classifier with a bimodal classifier. The tested modalities will be conceptual images and as the other modality, the titles for the concepts converted to images as discussed in 2.1. To classify images, a convolutional neural network is used, introduced as as a very successful method for image classification by Krizhevsky et al. (2012). The classification itself is performed by a fully connected layer. All neural implementations used Keras (Chollet et al. (2015)) working with the Theano library. In order to further design the neural network several parameters need to be tested and justified in order to create a working model. Most of the justification is found hereafter in section 2.2. For some parameters, experiments were performed which can be found in section 2.3. However, the construction of datasets is explained first.

2.1 Data construction

Originally, the dataset was intended to represent concepts in order to form clear conceptual feature spaces, relating to multiple modalities. As such, simple iconographic images are used, obtained from www.sclera.be under the creative commons license. We selected 250 images of these to represent a concept. To gain a larger and more useful set for the CNN, shear maps and elastic distortions were used to augment the dataset. Sclera icons mostly have black backgrounds and permit such transformations without introducing discontinuities. In total, 40 augmentations were used per class forming a dataset of 10250 images for 250 classes. Then, images were padded with the black background present in sclera icons so

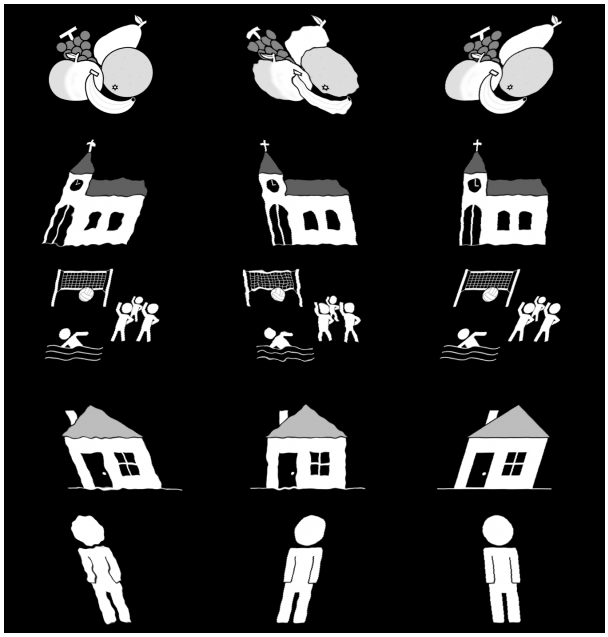

Figure 2.1: Sclera images used for visual representations of concepts. Each image is transformed 40 times using shear maps and random gaussian deformation.

Figure 2.2: Dataset on which the word classifier is trained. Each of 250 words are converted to images using 20 fonts.

that they share a fixed size. Images are finally reshaped to a resolution of 120 by 80 pixels. The transformations can be viewed in Fig. 2.1.

To provide another modality, words as images were obtained by converting the titles of every concept in the icon dataset, to images. Thus, every icon concept also has an image of its title. The images provide a different modality but use the same medium, factoring out spurious effects of using a different medium. Additionally, they also provide clear and representative information for any concept. For each title, 20 different fonts were used such that the dataset contains 20 variations for any title. The result was a dataset of 5000 images. Word-images here used a fixed size of 184 by 28 pixels, see Fig. 2.2.

2.2 Network design

Although neural classification networks do not require lots of layers to learn a correct solution, deep learning is generally favored over shallow models, especially for visual tasks. In Eigen, Rolfe, Fergus, and Lecun (2014) it was found that deeper convolutional models perform better than shallow models when limiting the number of model parameters, utilizing constraints on other parameters in the models. Moreover, in Ba and Caruana (2014), the authors showed that although shallow models are able to replicate the same functions as implemented by deep models, they generally are unable to learn that function from the provided data. Still, given the relatively simple task, 4 convolutional layers are used including 2x2 max-pooling on standard stride 2 after the first three convolutions.

In Ngiam et al. (2011), multimodal representations were encouraged by using at least one hidden layer, in contrast to a shallow model. Although the multimodal model already forms representations in the convolutional parts, an additional fully connected layer is appended to the convolutions and dubbed the representation layer. The representation layer also increases training speed by reducing the initial number of outputs to classify on. Perhaps more informative; The number of parameters grows multiplicatively between two fully connected layers and additively across them, so by introducing a smaller hidden representation some connections are substituted by the added depth. In order not to introduce bias between the unimodal and multimodal architecture, the same representation layer is used in the unimodal networks. A final classification layer is trained on top of the representation layer such that classification is based on the representations.

In convolutional neural networks, the rectified linear units (ReLU) supersede the alternative activation functions, e.g. hyperbolic tangent functions, as seen in Glorot et al. (2011). The ReLU's have a

constant gradient when positive, circumventing the vanishing gradient problem and add non-linearity to the network function. Given the former, rectified linear units are widely used in CNN's.

The kernel sizes were kept at 3x3, in principle with Szegedy, Vanhoucke, Ioffe, Shlens, and Wojna (2016) where the large inception network is researched. The paper proposes the idea to replace wide kernels with multiple convolutions using kernels of only 3x3, such that they stack up to have receptive fields at least the size of the original kernel.

All neural networks in this paper are trained using the adaptive optimiser adadelta, see Zeiler (2012). The adaptive optimiser provides good results without introducing more hyperparameters. Using other learning schedules, however, may be beneficial to accuracy after tuning. Given the already present hyperparameters and different networks used here, a choice was made to stick to adadelta. Not having to deal with the introduced parameters will prevent human error and reduces effects influencing the results.

Now that some global model parameters are determined, the number of kernels and nodes in the representation layer are determined using experiments. More information on how the experiments were set up follows after this section. The chosen parameters mentioned next will be used in the multimodal classifier. Now, for the sclera classifier, 6 different kernels are used on each convolution (half of that, 3 kernels, are used for the first layer in the unimodal networks) and 7 neurons on the representation layer. The total number of parameters were somewhat limited to allow for more training without reaching an accuracy above what we can safely evaluate using 1250 images in the test set. The image classifier has 9165 parameters in total. It was trained for up to 40 epochs using a batch size of 10 taking about 8 minutes. The processor used for training of all models was a quad-core (8 threads) 2.30 GHz CPU. Early stopping was applied whenever the accuracy did not increase by 0.5% for more than 2 epochs. The image classifier reached a top accuracy of 96.7% in 10 folds. For the word classifier 12 kernels in the same pattern as before and 55 neurons in the representation layer are used, resulting in 78111 trainable parameters. The size of the representation layer was chosen to have the word classifier match the accuracy of the image classifier. The CNN was trained for up to 150 epochs (although typically only 25 epochs are used) with a batch size of 10 on 4050 samples from the training set depending on the cross validation fold. With the smaller word data set, the network described here takes roughly 3 minutes to train. The parameters were determined on a 450 sample validation set and reached a max accuracy of 91.2% in 10 folds. Given the parameters above, the unimodal classifier for images is shown in Figure 2.3. The word classifier has similar structure but uses the corresponding parameters (12 kernels, 55 representation neurons).

In order to see the effect of multi-modality on performance in CNN's, a multi-modal neural network is needed. The main network uses the image classifier and word classifier, and replaces the two classification layers with a single retrained layer connected to both representations. The research question is whether the information gained by reading the words will increase the accuracy of prediction on seeing the concepts as images only, and vice versa. Thus, the multimodal architecture is compared to two control-condition networks. The comparison models forming the control-conditions will use exactly the same convolutional networks but have duplicate input CNN's. That is, they will be two image classifiers with a single shared classification layer, as well as a similar network using word classifiers. The control-conditions will indicate if any gained performance in the bigger network does not originate from the increase in architecture complexity but from the additional task information: the bimodality.

The multimodal model takes two classifiers, scraps the classification layer on both models and retrains a new shared layer. The new fully connected layer is trained for up to 150 epochs with early stopping on multimodal data. Early stopping was applied generally between 10-60 for the image classifier control-condition, 20-60 for the multimodal network and 40-80 for the comparison word architecture. The way data was fed into a multi input network is by providing one

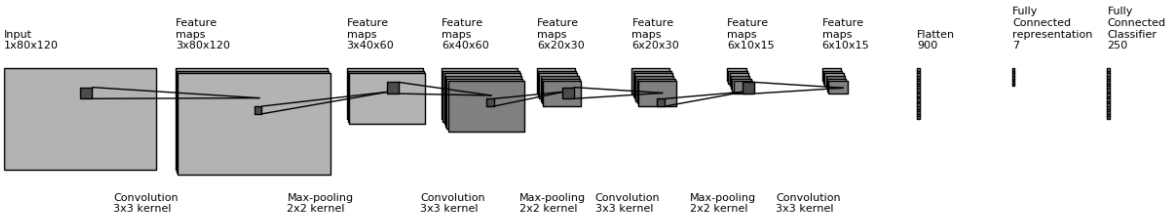

Figure 2.3: Convolutional neural network that classifies images into classes. The final fully connected layer is the classifier and is fixed at 250 neurons. The additional fully connected layer allows for representation which will be useful when using the multimodal architecture.

modality 30% of the time, the other modality also 30% and both modalities 40%. The multimodal architecture is shown in Figure 2.4. To understand the comparison architectures, imagine Figure 2.4 with the upper and lower neural networks equal to each other, using either the image, or word CNN design.

The CNNs are reused by removing the final classification layer for both inputs, and training a new multi-modal classifier on the remaining representation layers, see also Donahue, Jia, Vinyals, Hoffman, Zhang, Tzeng, and Darrell (2014) for more details on the operation. The same procedure is applied to the unimodal network to test overfitting in section 4. In order to reuse any CNN, features will have to transfer to the new neural network. Yosinski, Clune, Bengio, and Lipson (2014) talks about tranferability of features and found layers to co-adapt. Since the neural networks used in this paper are relatively shallow, little co-adaption should occur at all. Moreover, only the final layer is removed so no co-adaptation is lost either. Representation specificity with respect to the task should be no problem since the unimodal and multimodal data sets differ only in the folds used.

Using the design choices results in the architectures shown in Fig. 2.3 and 2.4. Note that specific numbers of kernels and neurons are determined hereafter, although the generic distribution where the first layers has half of the neurons of deeper layers is consistent.

2.3 Experiments

The optimal number of kernels is determined for sclera images by training the unimodal CNN over 2-20 kernels in step sizes of 2, keeping the rest of the model fixed with 7 nodes in the representation layer. To ensure proper results, the neural networks were trained using 10-fold stratified cross validation on 81% of the data. The experiment was validated on 9% of the dataset, depending on the fold. The remaining 10% will be used as a test set later on. The resulting accuracies for the image classifier as a function of kernels used can be viewed in Fig. 3.1. No such experiment was conducted for the word classifier after seeing that performance did not severely increase accuracy for more than 2 kernels. Instead, the number of kernels was determined intuitively. A similar effect of the ineffectiveness of feature maps on accuracy was found in Eigen et al. (2014).

A similar experiment is run to determine the optimal number of nodes in the representation layer, again performed on the image classifier. The same general approach is used using 10-fold stratified cross validation on 1-18 representation neurons with 6 kernels as determined to be sufficient from the previous experiment. The number of kernels, 6, was chosen to have good accuracy while still maintaining the ability to discern improvements given the size of the data and keeping in mind the number of parameters in the model. Still, there will be many parameters with respect to the data, but reducing it will prevent overfitting somewhat; More attention is given to overfitting in the discussion in section 4. The same testset as used for experiments on kernels was left out in

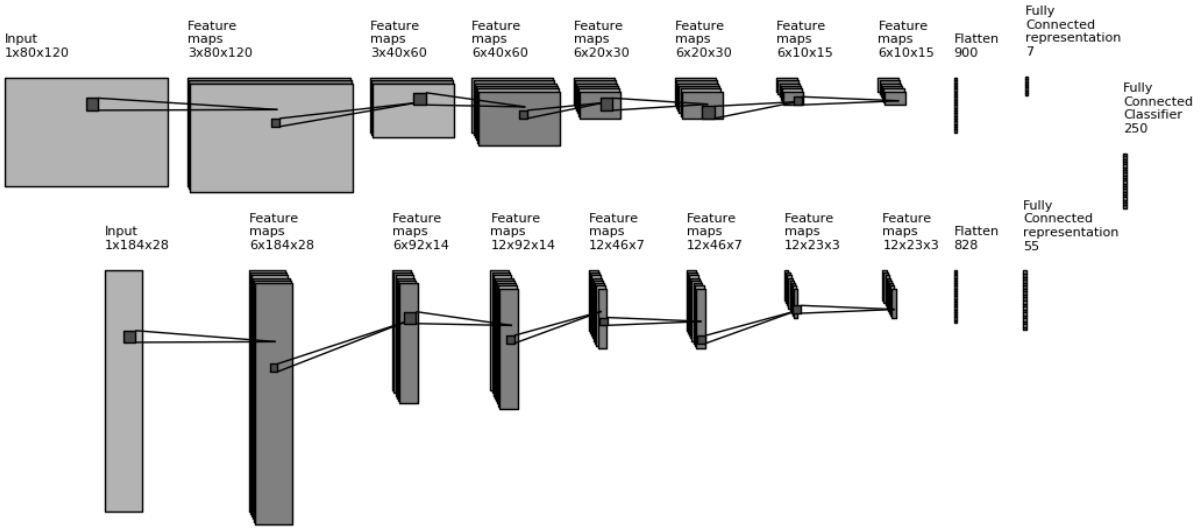

Figure 2.4: Convolutional neural network that classifies the inputs which may contain images and word images. The classification is performed on representations of both of the inputs.

training here. The results are shown in Fig. 3.2.

The number of neurons in the representation layer is correspondingly determined for the word classifier. This time, the number of neurons is validated between 3 and 66. Note that the word classifier requires much more neurons for reasonable classification. Results are shown in Figure 3.3. To get a similar accuracy as reached by the sclera classifier, 55 neurons were chosen to be in the representation layer.

The multimodel and comparison architectures are tested differently from how they were trained. That is, training involved providing both of the inputs generally but also leave one modality/input out sometimes to increase robustness. Additionally, voiding some inputs allows the neural network to also classify a sample when only one modality is provided. The procedure, then, permits three different test cases:

- Classify instances providing only input A
- Classify instances providing only input B
- Classify instances providing both inputs

By testing the inputs separately, one may discern whether the classification layer is, in fact, connected to both inputs at all, instead of only needing one input for correct classification.

3 Results

A convolutional neural network using the architecture in Fig. 2.3 is trained for classification for up to 40 epochs. The number of kernels on each but the first convolution varies between 2 and 20. The first layer has half as many kernels as the other layers which have as many as indicated. The results in Fig. 3.1 shows some increase of accuracy as the number of kernels are increased. A linear least squares regression ($0.882 + 0.005 * \#kernels$, $p < 0.0001$ on number of kernels) shows that the contribution of number of kernels is significant, mainly for low numbers of kernels. However, it is suspected that the number of kernels will show a similar curve to Figs. 3.2 and 3.3 with a high initial accuracy as it is bound to 100%. The contribution seems to quickly degrade in magnitude as the accuracy reaches top. To the same end, the regression line is left out so as not to suggest a linear relationship to the reader.

The same architecture is used, using 6 kernels but now varying the number of neurons in the representation layer from 1 to 18. From Fig.

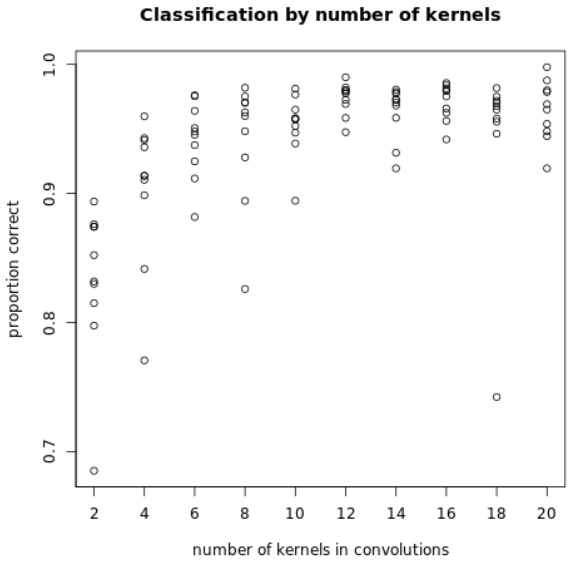

Figure 3.1: Classification accuracy by number of kernels used for each convolution. The first convolution always uses half as many kernels/filters as convolutions thereafter.

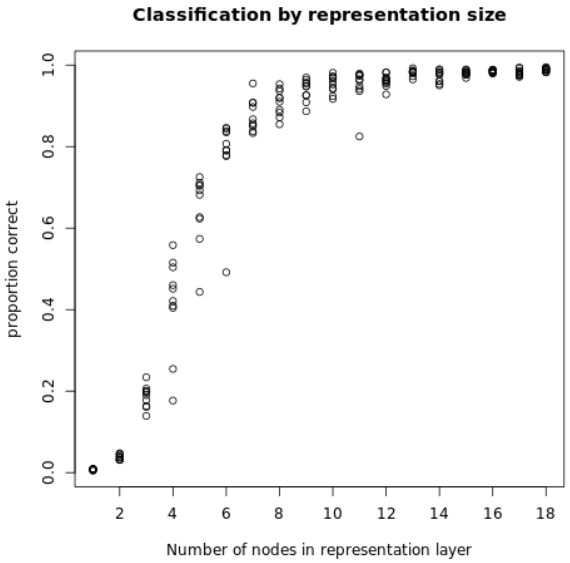

Figure 3.2: Classification accuracy for images, by number of neurons in the representation layer. The layer forms a bottleneck whenever the number of neurons is less than the 250 nodes in the classification layer.

3.2, it is observed that accuracy converges to a maximum as representation size grows larger in the shown range. It is possibly that with extraordinarily large representations, the number of parameters outgrow the training that the fixed size dataset can give, reducing the accuracy instead. For this, however, this research provides no results.

The word classification is a much harder task as seen from Fig. 3.3, where up to 66 nodes in the representation layer are shown.

The accuracies of multi-input neural networks are listed in Fig. 4.2. The results are notably worse than the unimodal classifiers, which is further analysed in the discussion and will include the interpretation.

4 Discussion

In Fig. 3.1 the accuracy mostly did not improve as a function of the number of kernels In section 3, a significant relation between accuracy and kernels is found, but relies mostly on the initial contributions. The ineffectiveness of larger number of kernels was also shown in Eigen et al. (2014). The number of kernels required was, therefore, determined in a more intuitive fashion. However, it would be interesting to test the effect of kernels in another way. For example, it is possible that more kernels increase the probability of successful learning. To test this effect, however, would require extensive looks to learning curves of CNN's which are not the main focus of this paper.

However, the accuracy shows a nice relationship with the number of nodes in the representation layer. The accuracy seems to grow slowly initially (see Fig. 3.2) then grow quickly after which the gradient goes back down (also Fig. 3.3). Intuitively, the initial increase in accuracy is because the representational strength is strongly bottlenecking the information. Hence, the accuracy rapidly goes up with the amount of unique combinations that the weights can make. The decreasing gradient can be because with a larger representational strength, the representation layer is no longer solely bottlenecking the CNN and competes with

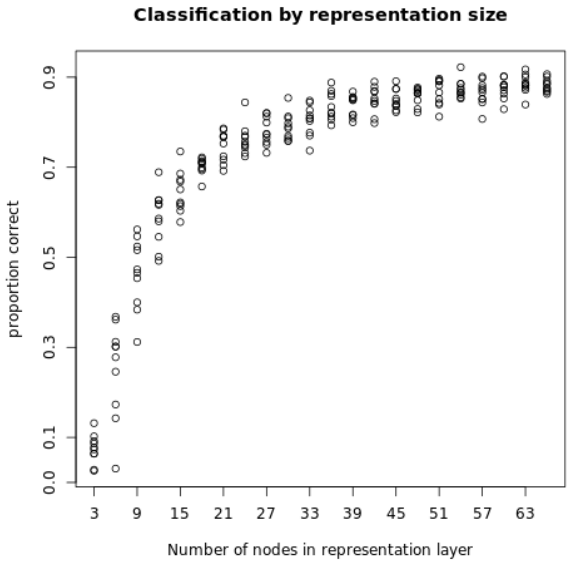

Figure 3.3: Classification accuracy for words, by number of neurons in the representation layer. The layer forms a bottleneck whenever the number of neurons is less than the 250 nodes in the classification layer.

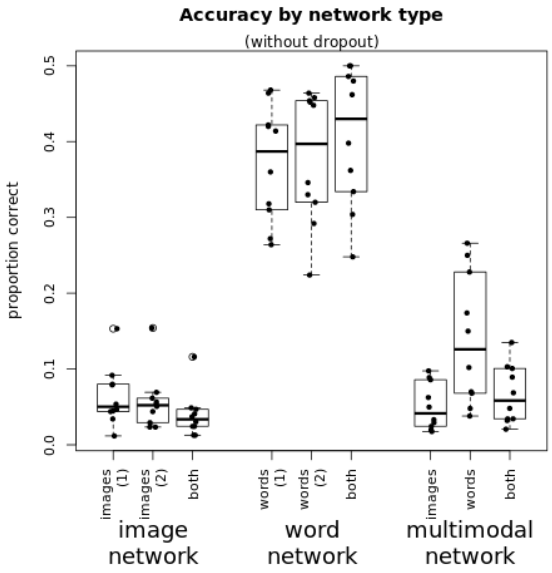

Figure 3.4: Proportion correct for the different multi input models, without dropout. Accuracies are exceptionally low for convolutional neural networks, analysed in sec. 4. See also table A.4 in the appendix for the data.

other architecture improvements for contributions. Since the representation size forms a bottleneck in the information flow, it is probably good to avoid such bottlenecks in an architecture. Similar remarks are made in Szegedy et al. (2016), opting for a smooth decrease of information through the neural network.

The results in Fig. 3.4 of the multimodal and control-condition models are very poor compared to most CNN's used in today's research, starting with Krizhevsky et al. (2012). The best CNN in Fig. 3.4 is the word classifier, which is also the most complex net. Both the image and multimodal classifier seem to not be able to learn a correct classifier. Since a simple architecture used for unimodal classification already performs better than the comparison architectures, something in the control setup already is detrimental to performance. In contrast to the unimodal networks which classify well, here, the classification layer was stripped from input CNN classifiers and retrained. Therefore, it seems that the retrained classification layer may be unable to learn from the provided representations. A possible explanation for the observed behaviour is that the representations provided are not useful or retraining a single layer on top is not an efficient way of learning. However, given the positive results of Donahue et al. (2014) under single layer retraining, the first focus will be inspecting the representations. The problem was found early on in the research, allowing more investigation into the problem. Seeing that the trained unimodal classifiers have good accuracy compared to the comparison architectures, despite representations being unable to transfer to the bimodal architectures, the problem is suspected to be overfitting of the representation on the unimodal data and setting. So, to test the first idea that representations are not useful, overfitting is combated using dropout (see Srivastava, Hinton, Krizhevsky, Sutskever, and Salakhutdinov (2014)), applied in the unimodal classifiers. Aside from the problems with overfitting which will be covered, Figure 3.4 shows maximum performance for word classification. It is expected that, given the problem of overfitting, the word classifiers provide most support for recovery with 55 neurons in the

representation layer.

4.1 Representation by dropout

To test whether overfitting is the problem, the image classifier is initially trained with and without dropout. Parameters for the models are the same as the original unimodal networks, however, early stopping was applied when accuracy did not improve for more than 2 epochs by 0.05% with a maximum of 40 epochs. Dropout in a unimodal classifier is applied throughout the neural network with 20% dropout on the input, 10% dropout after convolutions and 50% on fully connected layers, much in line with the original paper by Srivastava et al. (2014). Since now only about half of the neurons are activated when using dropout, using the method requires increasing the width of some layers. For the unimodal classifiers, the representation layer was doubled in size (14 for images and 110 nodes for word classification).

Next, the classification layer is stripped from the classifiers with and without dropout (The representation layer remains intact). After stripping the classification layer, a new classification layer is trained on the same architecture, freezing all other weights in the process. By retraining the classification layer only, the process used for training the bimodal and comparison networks is simulated. Testing the simulated neural networks with dropout on the image testset, results in the accuracies in Figure 4.1. The first and third plots in Figure 4.1 show that retraining the classification layer destroys much of the original network performance. Hence, the new classification layer is unable to learn from the model without dropout. When using dropout in the second and fourth plot (Fig. 4.1), conversely, the final performance remains almost unharmed. Since dropout prevents overfitting and promotes individual neuron contributions, the extreme performance drop is due to the overfit representation neurons. Mainly, neurons in the unimodal classification layer have relied on an overfit representation layer, which only helps in the unimodal setting. The resulting connections after the representation layer have little real meaning and provide no basis for further training. Additionally, the observed loss of accuracy from classifier retraining shows that the

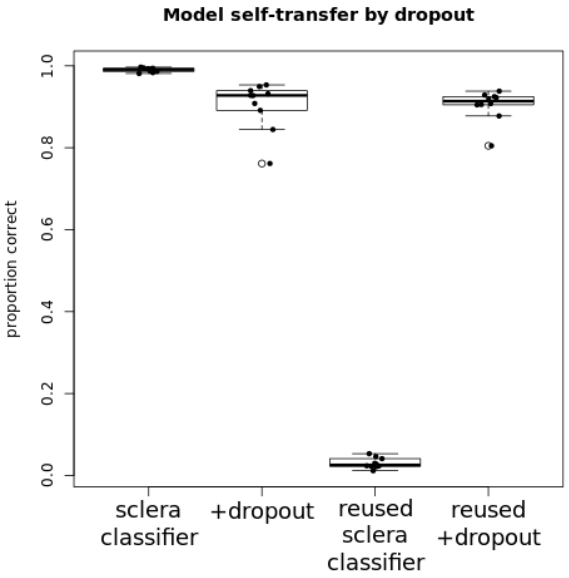

Figure 4.1: Classification accuracy on a unimodal image classifier in plots 1 and 3, and on a similar neural network with dropout for the second and fourth plots. The third and fourth graphs have their classification layer reset and retrained with other weights frozen. The classifier is able to recover when using dropout. See also table A.5 in the appendix for the data.

representation specificity relevant in Yosinski et al. (2014) also includes overfitting on data. The result discussed here prompts a repeat of the multimodal experiment, but using dropout. One note needs to be added to the former. Since the problem of overfitting was found early on in validation folds, the first results on multimodal networks were using dropout. The experiments without dropout were then recreated after the experiment with dropout. The results are, thus, reversed with respect to the order depicted in this paper for reasons of clarity.

The multimodal experiment is repeated using unimodal classifiers with dropout as inputs to a bimodal layer, trained as described earlier in this section. The results of the multimodal experiment with dropout are shown in Figure 4.2. All neural networks have improved on the results in Figure. 3.4 and seem to learn classification well. That is, the comparison networks for image and word classification as well as the multimodal classifier have

improved upon their counterpart without dropout. To test for significance, Wilcoxon rank sum tests are performed for each type of neural network with dropout given both inputs and compared to the net without dropout. However, the relatively small testset for the multimodal input results in some tied accuracies. Using different rankings for these tied accuracies in 10 tests the p-value was found to be < 0.0001 consistently between the neural networks with/without dropout, given both inputs. Thus, dropout significantly improves the classification of the multi input networks.

More strongly, the multimodal architecture improves on the comparison architectures. To test for significance, Wilcoxon rank sum tests are performed between the multimodal network and the multi-input image and word network. With the same accuracy ties, again multiple rankings were done. In 10 different rankings the p-value was found to be < 0.0001 with the same p-value consistently, for the multimodal net against the image classifier. Between the multimodal and word networks, again a consistent p-value is found < 0.0001. The increase in accuracy for a multimodal network over the comparison architectures positively answers the research question. That is, adding different media for the input data increases the performance of a convolutional neural classifier, more so than increasing the complexity of the classifier itself as shown by doubling the unimodal architectures in a similar way. As a matter of fact, the comparison architecture for words is more complex than the bimodal classifier, without similar gain in accuracy.

Another observation in Figure. 4.2 is that the comparison architectures seem to also benefit from the increased complexity. More specifically, they benefit from using both inputs. However, the resulting accuracy given only one input is lower than the accuracy of a similar but unimodal classifier. Thus, the performance for a single input actually has gone down with respect to unimodal classification. The reduced accuracy, therefore, indicates the adverse effect of training on one input for only 30% of the time, instead of throughout.

It is also important to know whether the bimodal classification layer correctly connects to both inputs. If only one input is used, the

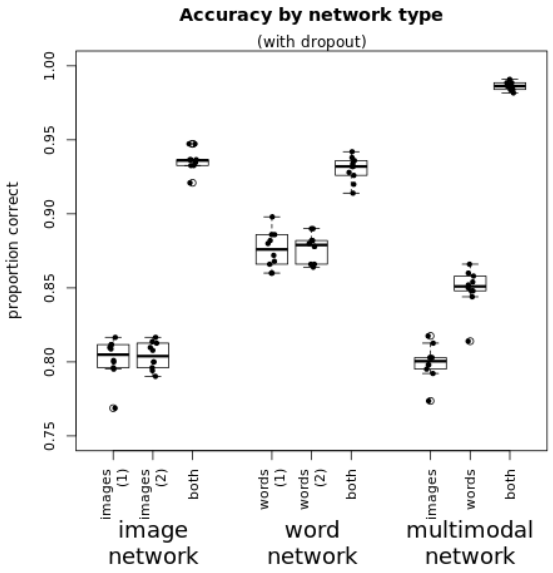

Figure 4.2: Proportion correct for the different multi input models, with dropout. The multimodal network is found to have significantly higher accuracy (with $p < 0.0001$) than the control conditions for image and word classification. The three models have improved accuracy with respect to the same models without dropout in Figure. 3.4, given both inputs. See also table A.6 in the appendix for data.

accuracies for the corresponding model will be asymmetric between inputs. Because the accuracy of the control-condition architectures is very similar for one input to the other, the layer retraining seems to be correctly connecting to both inputs. The multimodal network does show asymmetry. However, given that the representations for word classification are more useful than those for image classification as seen from the respective single input accuracy differences in Figureure 3.4, the asymmetry is better explained by the increased usefulness of word representations.

It is possible that the main results of Figure 4.2 is due to unforeseen factors. For example, Fig. 4.2 does not show how useful the representations in the neural networks are. Thus, it is imaginable that the greater multimodal performance is only due to increased generalisation to new data. Then,

the performance increase by multimodality may depend on how good the existing neural networks already were at generalising. In that case applying multimodality in maximally generalising neural networks would not increase the performance. However, the applied dropout increases generalisation and reduces any effects of increased generalisation due to multimodality. A stronger test using maximally generalising networks will be able to negate the factors mentioned here.

Another alternative explanation would assume not multimodality, but weight differences to be sufficient for the observed increase of accuracy. In that case the increase is because the multimodal classifier has learned to utilise the differences between multimodal inputs, contrary to the comparison architectures which use copy inputs. Redoing the experiment with the two best unimodal classifiers instead of copies would confirm beliefs. Until such confirmation, the weights differences are assumed not to hold enough relevant information for the observed improvement. More informatively, if a classifier is optimally generalising, then representations contained in the weights are assumed to be similar for a different iteration of it. Combining different classifiers as inputs to the comparison network, then, adds ample information to the multi-input classifier.

For further and similar research it is recommended to have a structured and ordered setup of experiments, including easily accessible validation sets. Spurious effects in the multi-case experiments required careful control and lots of testing. Being able to validate approaches frequently will allow more ideas to be tested, and improve the eventual result.

Other suggestions and future considerations apply to the retrained layers as used in the bimodal and comparison networks. For these bimodal networks a new classification layer needs to be retrained using transfer learning methods. In this paper, a single layer was retrained. Yet, Yosinski et al. (2014) find that fine-tuning after retraining a layer increases generalisation. It is, therefore, better to use finetuning when transferring networks.

A final interesting opportunity opens up in the feature space of multimodal networks. This paper show an increase of accuracy for multimodality in CNN's, but to understand why, it can be very interesting to see the effect on samples in the feature space. For example, given easier classification of multimodal data may indicate a decrease in ambiguous samples in feature space. Observing such a multimodal conceptual feature space may even provide insight into the value of related spaces.

References

Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2654–2662. Curran Associates, Inc., 2014. URL `http://papers.nips.cc/paper/5484-do-deep-nets-really-need-to-be-deep.pdf`.

Grard. Bailly, Pascal Perrier, and Eric Vatikiotis-Bateson. *Audiovisual Speech Processing*. Cambridge University Press, 2012. doi: 10.1017/CBO9780511843891.

Samy Bengio. Multimodal speech processing using asynchronous hidden markov models. *Information Fusion*, 5:81–89, 2004.

Lynne Bernstein and Christian Benoît. For speech perception by humans or machines, three senses are better than one. *Proceedings of the International Conference on Spoken Language Processing*, 3:1477–1480, 12 1996.

François Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, pages 647–655. JMLR.org, 2014. URL `http://dl.acm.org/citation.cfm?id=3044805.3044879`.

Susan T. Dumais. Latent semantic analysis. *Annual Review of Information Science and Technology*, 38:188–230, 2004.

David Eigen, Jason Rolfe, Robert Fergus, and Yann Lecun. Understanding deep architectures using a recursive convolutional network. In *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*, 2014.

Andrea Frome, Greg Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. *Neural Information Processing Systems*, 2013.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. *Journal of Machine Learning Research*, 15:315–323, 2011.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. URL `http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf`.

Geoffrey E. Hinton, Simon. Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. neural computation. *Neural Computation*, 18:1527–1554, 2006.

Jing Huang and Brian Kingsbury. Audio-visual deep learning for noise robust speech recognition. *International Conference on Acoustics, Speech and Signal Processing*, pages 7596–7599, 10 2013. URL `https://doi.org/10.1109/ICASSP.2013.6639140`.

P. Diederik Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations*, 2014.

Alex Krizhevsky, Ilyra Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, 2012.

Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv:1512.09300*, 2016.

Harry McGurk and John MacDonald. Hearing lips and seeing voices. *Nature*, 264:746–8, 12 1976.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations. In *Workshop International Conference on Learning Representations*, 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013b. URL `http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf`.

Youssef Mroueh, E Marcheret, and Vaibhava Goel. Deep multimodal learning for audio-visual speech recognition. *International Conference on Acoustics, Speech and Signal Processing*, pages 2130–2134, 04 2015. URL `https://doi.org/10.1109/ICASSP.2015.7178347`.

Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. NG. Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine learning.*, 2011.

Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg Corrado, and Jeffrey Dean. Zero-shot learning by convex combination of semantic embeddings. *Computer Science*, 2014.

Lawrence D. Rosenblum. Speech perception as a multimodal phenomenon. *Current Directions in Psychological Science*, 17(6):405–409, 2008a. URL `http://dx.doi.org/10.1111/j.1467-8721.2008.00615.x`. PMID: 23914077.

Lawrence D. Rosenblum. *The Handbook of Speech Perception*, chapter Primacy of Multimodal Speech Perception,

pages 51–78. Blackwell Publishing Ltd, 2008b. ISBN 9780470757024. URL http://dx.doi.org/10.1002/9780470757024.ch3.

L. Schomaker, J. Nijtmans, A. Camurri, P. Morasso, C. Benoit, T. Guiard-Marigny, B. Le Gof, J. Robert-Ribes, A. Adjoudani, I. Defee, S. Munch, K. Hartung, and J. Blauert. A taxonomy of multimodal interaction in the human information processing system: Report of the esprit project 8579 miami. Technical report, Nijmegen University, NICI, 1995.

Paul Smolensky. Information processing in dynamical systems: Foundations of harmony theory. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, 1:194–281, 1986.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems 27*, pages 3320–3328, December 2014.

Matthew D. Zeiler. Adadelta: an adaptive learning rate method. *arXiv:1212.5701*, 2012.

A Supporting tables

Table A.1: Average proportions for Fig. 3.1, image classification using 7 representation neurons by number of kernels used in each but the first layer.

# kernels	2	4	6	8	10	12	14	16	18	20
proportion correct	0.833	0.903	0.942	0.942	0.953	0.974	0.963	0.971	0.943	0.964

Table A.2: Average proportions for Fig. 3.2, image classification using 6 kernels by number of nodes in the representation layer.

representation neurons	1	2	3	4	5	6	7	8	9
proportion correct	0.008	0.038	0.187	0.416	0.649	0.780	0.877	0.908	0.939
representation neurons	10	11	12	13	14	15	16	17	18
proportion correct	0.953	0.948	0.961	0.981	0.973	0.981	0.985	0.981	0.989

Table A.3: Average proportions for Fig. 3.3, word classification using 12 kernels by number of nodes in the representation layer.

representation neurons	3	6	9	12	15	18	21	24	27	30	33
proportion correct	0.075	0.252	0.464	0.588	0.645	0.703	0.747	0.762	0.780	0.790	0.804
representation neurons	36	39	42	45	48	51	54	57	60	63	66
proportion correct	0.836	0.837	0.849	0.850	0.859	0.867	0.874	0.865	0.874	0.885	0.882

Table A.4: Data for Fig. 3.4, proportions correct by multimodality and control networks without dropout. Second rows mark the type of input used and the last row shows the means.

Image classifier			Word classifier			Multimodal classifier		
Images 1	Images 2	Both	Words 1	Words 2	Both	Images	Words	Both
0.079	0.054	0.030	0.272	0.292	0.304	0.020	0.070	0.032
0.045	0.023	0.037	0.464	0.458	0.480	0.086	0.266	0.135
0.153	0.154	0.116	0.468	0.452	0.500	0.098	0.174	0.103
0.080	0.057	0.049	0.360	0.346	0.398	0.062	0.250	0.089
0.044	0.029	0.024	0.318	0.320	0.362	0.089	0.228	0.101
0.092	0.061	0.041	0.420	0.454	0.486	0.018	0.048	0.034
0.047	0.051	0.047	0.422	0.448	0.500	0.050	0.150	0.067
0.034	0.023	0.013	0.310	0.330	0.334	0.029	0.038	0.021
0.054	0.044	0.024	0.414	0.464	0.462	0.024	0.068	0.034
0.012	0.069	0.013	0.264	0.224	0.248	0.033	0.102	0.048
0.064	0.057	0.039	0.371	0.379	0.407	0.051	0.139	0.067

Table A.5: Proportion correct for each iteration corresponding to Fig. 4.1; Observing the effect of dropout on self-transfer learning. *+Dropout* indicates the same image classifier with dropout applied. *Reused* signifies the classification layer is retrained. Means are shown at the right end.

sclera classifier	0.987	0.993	0.994	0.995	0.992	0.986	0.997	0.983	0.981	0.990	0.990
+dropout	0.949	0.932	0.762	0.953	0.845	0.908	0.939	0.928	0.928	0.891	0.903
reused sclera classifier	0.028	0.041	0.046	0.022	0.024	0.023	0.030	0.053	0.021	0.012	0.030
reused +dropout	0.922	0.925	0.805	0.938	0.878	0.905	0.919	0.908	0.929	0.905	0.903

Table A.6: Data for Fig. 4.2, proportions correct by multimodality and control networks with dropout. Second rows mark the type of input used and the last row shows the means.

Image classifier			Word classifier			Multimodal classifier		
Images 1	Images 2	Both	Words 1	Words 2	Both	Images	Words	Both
0.812	0.796	0.933	0.868	0.864	0.928	0.795	0.866	0.986
0.801	0.810	0.947	0.860	0.866	0.914	0.792	0.850	0.986
0.795	0.794	0.937	0.898	0.890	0.938	0.798	0.848	0.984
0.817	0.800	0.936	0.882	0.878	0.932	0.818	0.814	0.982
0.769	0.800	0.921	0.886	0.890	0.932	0.798	0.854	0.984
0.810	0.808	0.937	0.860	0.880	0.926	0.803	0.860	0.991
0.812	0.814	0.933	0.866	0.882	0.942	0.813	0.844	0.984
0.800	0.813	0.937	0.872	0.882	0.934	0.774	0.858	0.989
0.796	0.790	0.947	0.880	0.866	0.920	0.803	0.852	0.986
0.809	0.817	0.935	0.886	0.878	0.936	0.803	0.848	0.989
0.802	0.804	0.936	0.876	0.878	0.930	0.800	0.849	0.986