Ben-Gurion University of the Negev
Faculty of Engineering Sciences
Department of Software and Information systems Engineering

Deep Learning
Assignment 2

Sol Amara
Ido Gurevich

Introduction

The dataset consists of 13,233 images of 5,749 different individuals. The training data includes 2,200 pairs of photos, with 1,100 pairs of images of the same person and 1,100 pairs of different people. Similarly, the test data consists of 1,000 pairs, evenly split . All images in this dataset are greyscale with dimensions of 250x250 pixels.
We split the training dataset into training and validation sets, maintaining balance between pairs from the same person and pairs from different people. The training set includes 886 pairs from the same person and 880 pairs from different people, while the validation set includes 212 pairs from the same person and 220 pairs from different people. Additionally, to ensure effective validation, we made sure that in the validation set and in the training set have different individuals' images. Meaning that if an individual person appeared in the training set, any of this specific individual will appear in the validation set.

As part of the preprocessing, we normalized the pixel values (with a range of [0,1]) and resized the photos to 105x105 pixels. These modifications were made to reduce computation time by handling smaller data and to align with the specifications of the referenced paper.

For model training, we utilized the Siamese CNN architecture proposed in the paper (Fig. 1). The training was conducted with a batch size of 100 for 30 epochs, with the learning rate set to 10^{-4} as specified in the reference paper.

The stopping criteria combined two different conditions:

1. The first condition goal is to ensure that the model is learns the training dataset and combined that:
   1.1 The training set accuracy exceed 95%
   1.2 The mean improvement in train set loss over the last 6 epochs is lower than 0.01, meaning that the model stops learning.

Or

2. The second condition goal is to avoid overfitting (meaning that the model keeps learning the training set but can't do generalization to new data) and combined that:
   2.1 The 'slope' of the validation loss (calculate by the different between two points) is higher than 0.05
   2.2 The validation loss is lower than the train loss

The Binary Cross-Entropy (BCE) loss function is used to measure the difference between predicted and actual class probabilities, making it ideal for binary classification tasks like determining whether photo pairs belong to the same person or different people.

We used the Adam optimizer due to its efficient handling of sparse gradients and adaptive learning rate, which facilitates faster convergence and improved performance on complex datasets.

<u>Models:</u>

1. CNN

This model has the same architecture (figure 1) as presented in the paper.

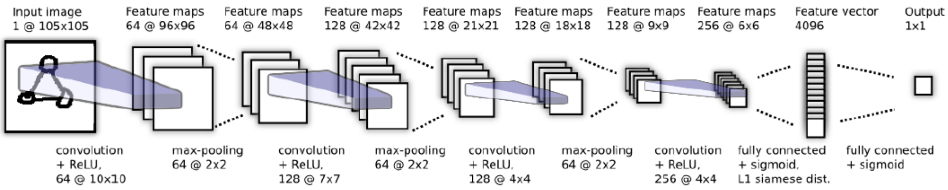

Figure 1: Network property

We observe the accuracy and loss trends on both the validation and training sets (Fig. 2). High training accuracy indicates the model fits well to the training data but struggles to generalize to unseen data, as shown by the lower validation accuracy, indicating overfitting. Additionally, the training loss decreases with more epochs, while the validation loss stagnates, further evidencing overfitting despite early stopping.

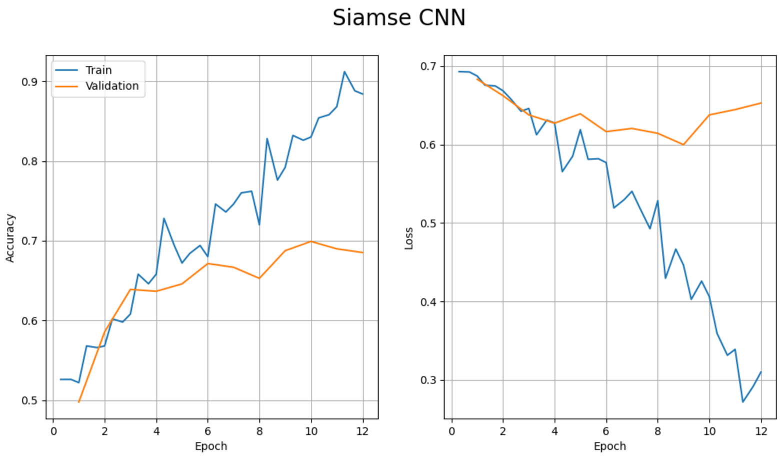

Figure 2: Accuracy and loss graphs

2. Siamse CNN+ Weight decay

Since our first model suffered from overfitting we tried to overcome this problem by adding weight decay. Weight decay is a regularization technique used to prevent overfitting. It works by adding a penalty term to the gradient update which effectively penalizes large weights during the optimization process.

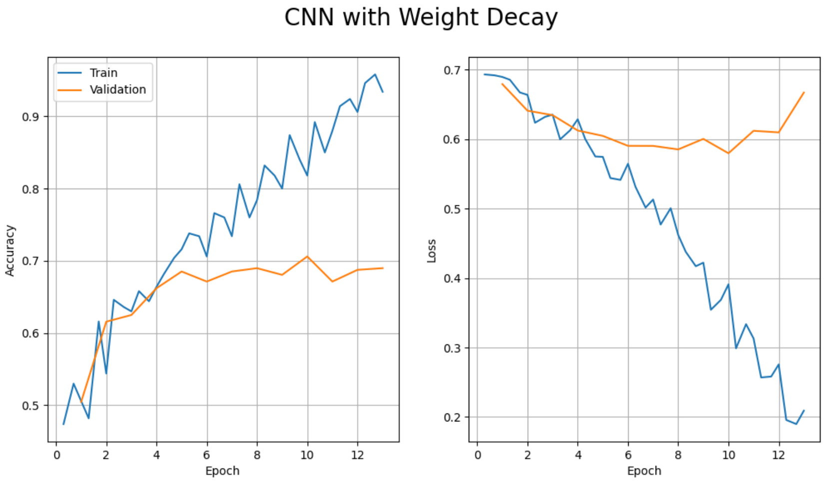

Figure 3: Accuracy and loss graphs

As illustrated in Fig. 3, we can see the accuracy and loss curves on the training set and validation set have the same trend until epoch 6, where the accuracy increases while the loss decreases. After epoch 6, on the validation set, the trends become different. This suggests that we still have the same problem as in the first model. When comparing the trends of the accuracy and loss curves of the two models in Figures 2 and 3, we can't see that using weight decay improves the performance of the models, as the trends are very similar to each other. However, Table 1 shows a slight improvement, with about 2% higher accuracy on the test set.

3. Siamse CNN+ Weight decay + Dropout

Here, we tried different approach in order to fight the overffiting, we chose to utilize dropout. When using dropout during training, randomly selected neurons are ignored, meaning they don't contribute to the forward pass. This forces the network to learn redundant representations of the data and improves its generalization ability. The dropout layers added after first 3 max pooling layers with value of 0.3 meaning that during training, 30% of the neurons in each dropout layer are randomly set to zero.

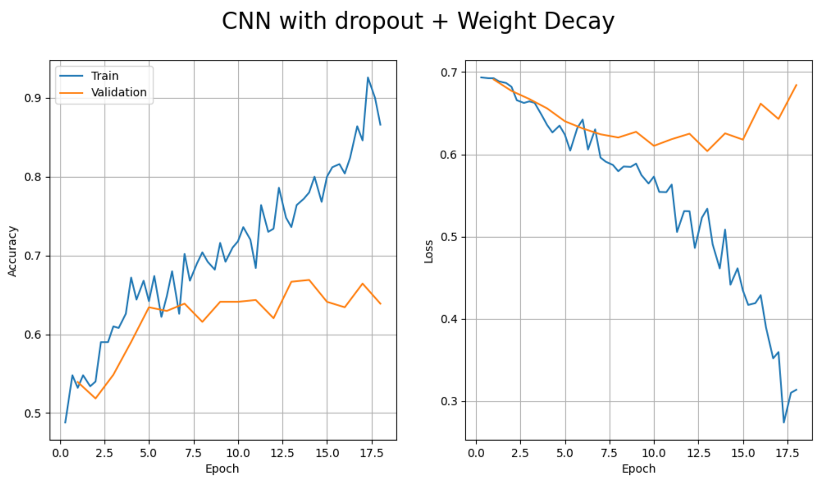

Figure 4: Accuracy and loss graphs

As observed in the previous model and as expected, the graph shows that the accuracy of the training set increases while its loss decreases. However, on the validation set, the loss begins to increase after approximately 12 epochs, indicating that the model struggles to generalize well. Despite a slight decrease in the final accuracy on the training set, there are minimal changes in the final accuracy on the validation and test sets, suggesting that the model is improving its ability to learn from new data, albeit at the cost of performance on the training data.

4. More channels

This model shares the same architecture as the dropout model but with a few modifications: we increased the number of channels per layer in layers 1 to 3. Additionally, we used the L2 function as the similarity function instead of L1. Changes illustrate in figure 5.

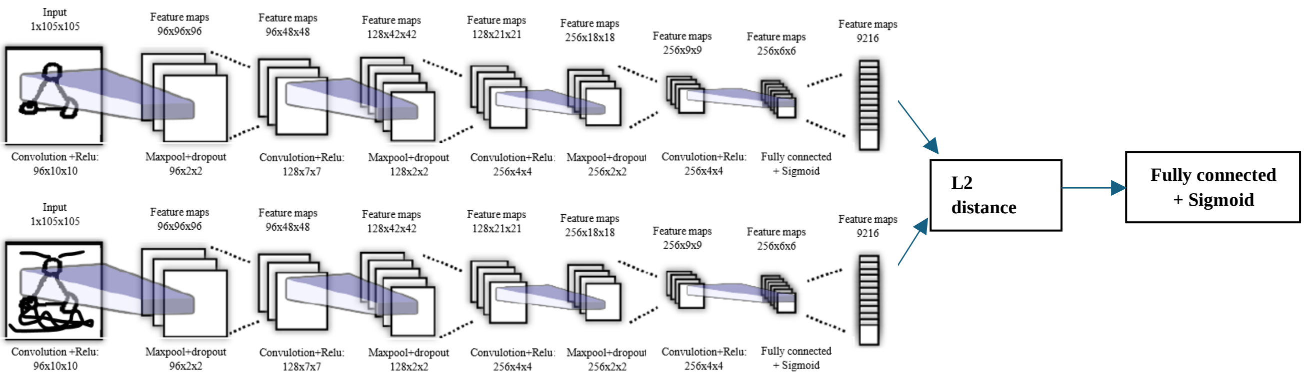

Figure 5: Network's architecture

This method improves upon previous results, as evidenced by higher final validation and test accuracies. Furthermore, these accuracies indicate a closer alignment between training and validation/test performance compared to previous models. The smaller gap between them suggests that the model may generalize better and be less prone to fitting than in previous scenarios with wider discrepancies.

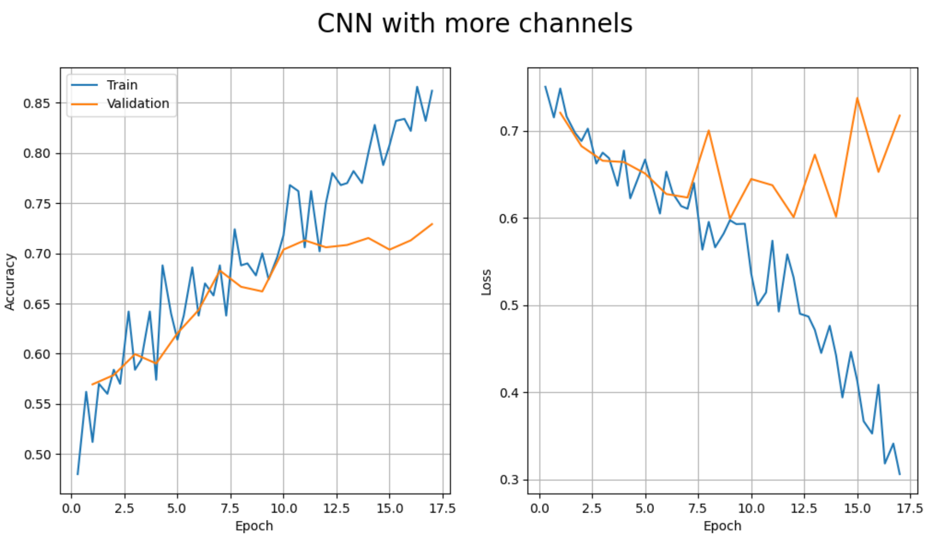

Figure 6: Accuracy and loss graphs

Evaluation

Table 1: Models performances summary

	Model	Convergence times [sec]	Final train loss	Final validation loss	Accuracy on train	Accuracy on validation	Accuracy on test
1.	**Siamse CNN**	**1288**	**0.3**	**0.65**	**0.88**	**0.68**	**0.643**
2.	**Siamse CNN + Weight decay**	**1344**	**0.21**	**0.66**	**0.934**	**0.689**	**0.663**
3.	**Siamse CNN + Weight decay + Drop out**	**1953**	**0.31**	**0.684**	**0.865**	**0.638**	**0.675**
4.	**More channels**	**2731**	**0.29**	**0.76**	**0.8**	**0.71**	**0.689**

We can see, according to Table 1, that the model with more channels, marked in red, outperformed the others. This model achieved the highest accuracy on both the validation and training sets. Additionally, it took longer to converge or to overfit compared to the other models. This occurred because increasing the number of channels means more parameters to learn. Moreover, the first three models have similar performance, with the main difference being in running time. The running time of the third model was about 600 seconds (10 minutes) longer compared to the first two models, indicating that using dropout slowed the process of overfitting.

Example of accurate classification:

Figure 7: Accurate classification

Example of misclassifications:

Figure 8: Misclassification

In the first pair of images, their facial features are very unique and similar, such as the mustache and hairstyle, making it easier for the algorithm to classify them correctly as the same person. In contrast, the algorithm misclassifies the second pair. It can be seen that in this pair, there are some differences making it challenging to classify, such as the lighting, position of the picture, facial expression, and the hand hiding part of the face. These factors create a challenging scenario for the model. They introduce noise and reduce the consistency of identifiable features, leading to a higher likelihood of misclassification. This highlights the importance of training models on diverse datasets that include various lighting conditions, angles, expressions, and obstructions to improve their robustness and accuracy in real-world applications.

Discussion

Based on the experimental results with different variations of the Siamese CNN architecture, several insights can be drawn. The initial Siamese CNN model, as depicted in Figure 1 and trained without additional regularization techniques, achieved reasonable training accuracy but struggled with validation and test sets, indicating potential overfitting. Introducing weight decay in the second model iteration improved training set performance slightly, yet the validation accuracy plateaued early, suggesting limited generalization. The third model, incorporating dropout alongside weight decay, showed improved generalization potential, albeit with a fluctuating validation loss after a certain point, hinting at a delicate balance in regularization effectiveness. Finally, increasing the number of channels yielded the highest validation and test accuracies, indicating enhanced generalization without significant sacrifice in training set accuracy.

In summary, while each enhancement—weight decay, dropout, and more channels—addressed different aspects of model robustness and generalization, they also introduced trade-offs in performance metrics. The final model with more channels demonstrated the best compromise, showing improved generalization capabilities across validation and test datasets while maintaining competitive training accuracy. These findings underscore the importance of balancing model complexity with regularization techniques to achieve optimal performance in facial recognition tasks.