Assignment 2 - DL

[redacted]

Analysis of the dataset

Number of pairs in each class in the dataset (equal/ not equal pairs)

	Train	**Test**
Equal	1100	500
Not equal	1100	500
Total pairs	2200	1000

Full experimental setup

Parameter description	**Parameter value**
batch size	100
Number of epochs	100
Regularization	L2

We compared between two different architectures, the first is the architecture presented in the paper (without batch normalization layers) and the second architecture has the same parameters structure with batch normalization layers.

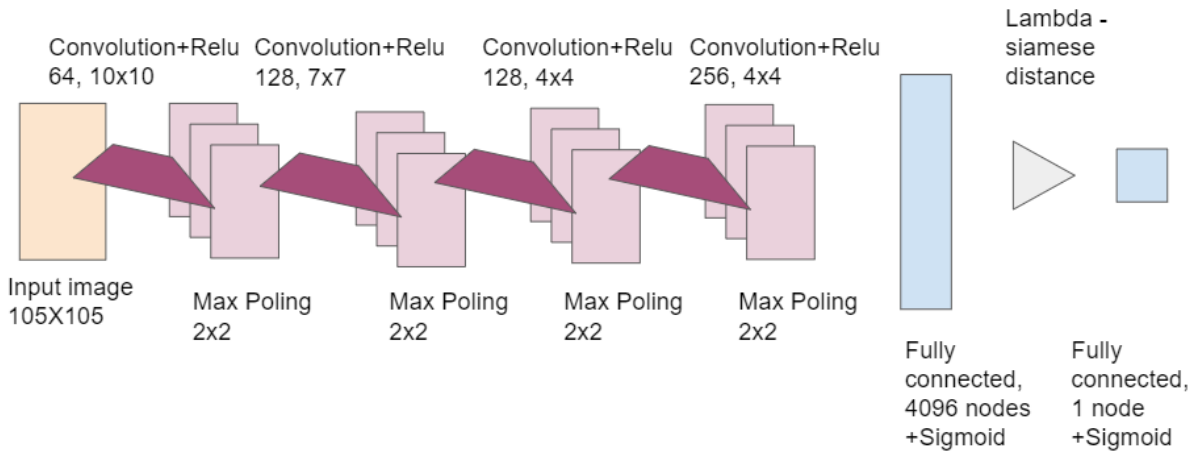
The hyper parameters we considered were for the learning rates and optimizers:
Learning rates = [0.0005, 5e-05, 5e-06]
optimizers = ["nadam", "adam"]

We used an early stopping callback- the training was stopped if the loss on the validation set didn't change for an entire epoch.

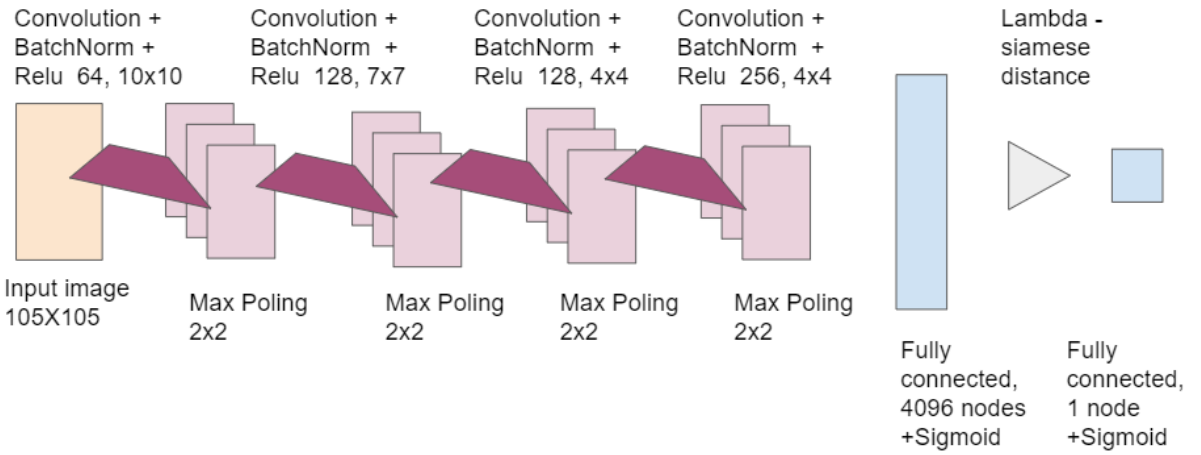
The loss function was binary cross entropy.
We used L2 regularization in all layers except the final layer (fully connected, 1 node).
We used the L2 regularization because it was used in the paper as well.

We initialized all network weights in the convolutional layers from a normal distribution with zero-mean and a standard deviation of 10−2. Biases were also initialized from a normal distribution, but with mean 0.5 and standard deviation 10−2.

First architecture design - without Batch Normalization layers:



Second architecture design - with Batch Normalization layers:



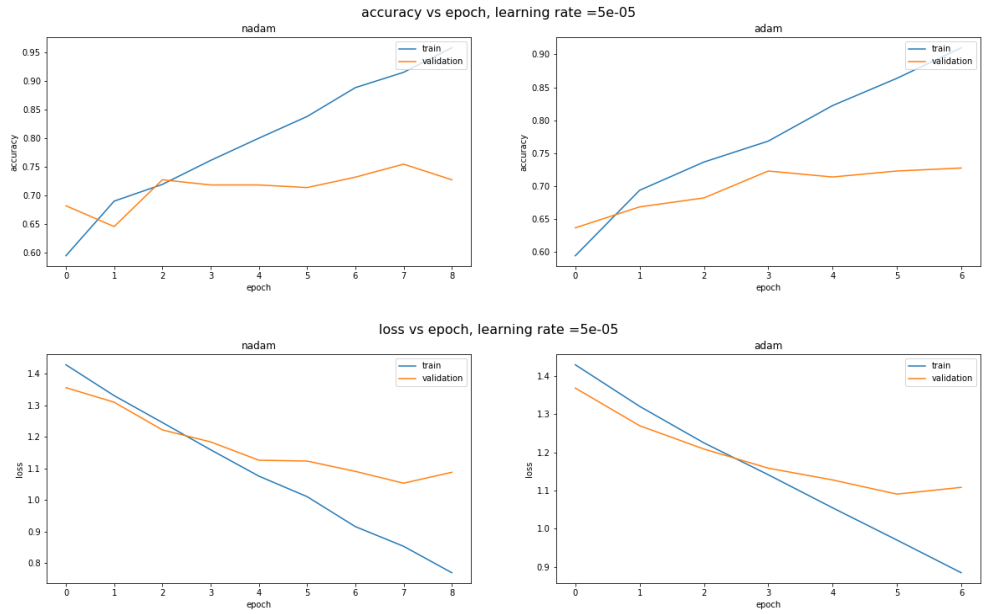
We evaluated the different architectures for all optimizers and learning rates we considered, we will present all the results and present what was our best model.

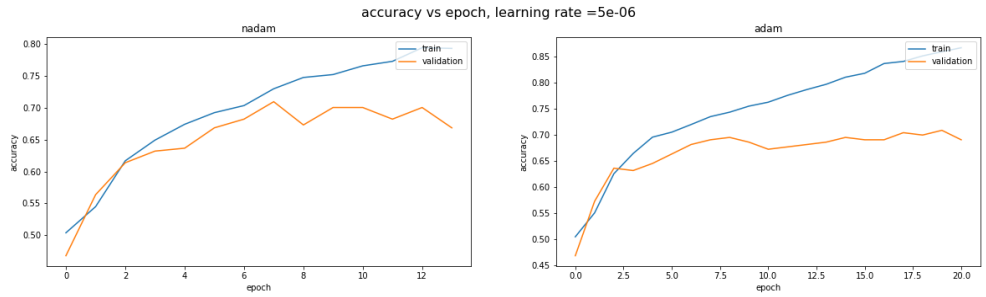
Architectures performance:

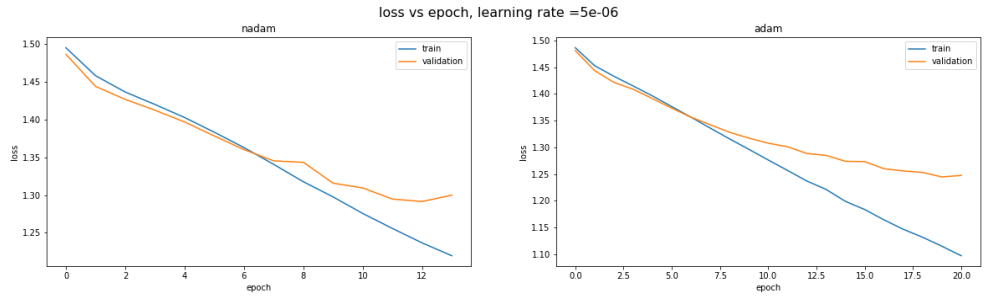
First architecture design - without Batch Normalization layers:

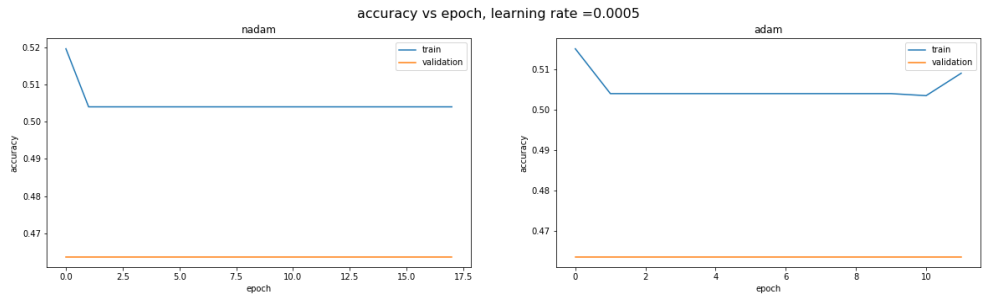
hyper parameters	Validation final loss	Validation Accuracy	Test final loss	Test Accuracy	Converge epoch	Training time
0.0005 (lr), ‘nadam’	0.745	0.463	0.733	0.5	18	158.252
0.0005 (lr), ‘adam’	0.784	0.463	0.771	0.5	12	89.644
5e-05 (lr), ‘nadam’	1.087	0.727	1.131	0.707	9	76.643
5e-05 (lr), ‘adam’	**1.108**	**0.727**	**1.137**	**0.720**	**7**	**55.039**
5e-06 (lr), ‘nadam’	1.299	0.668	1.301	0.705	14	116.976
5e-06 (lr), ‘adam’	1.247	0.690	1.267	0.708	21	157.803

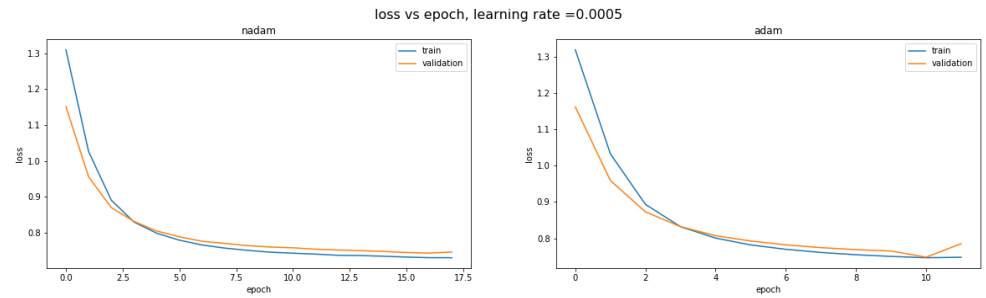
Performance figures (for each learning rate and optimizer):











The best model with the first architecture is the model with the hyper parameters: 5e-05 (lr), adam (optimizer) since it had the best accuracy result on the validation set and it converged fastest (7 epochs) .We will present this model classification matrix and classification examples:

		True class	
		Equal	Not equal
Predicted class	Equal	391	171
	Not equal	109	329

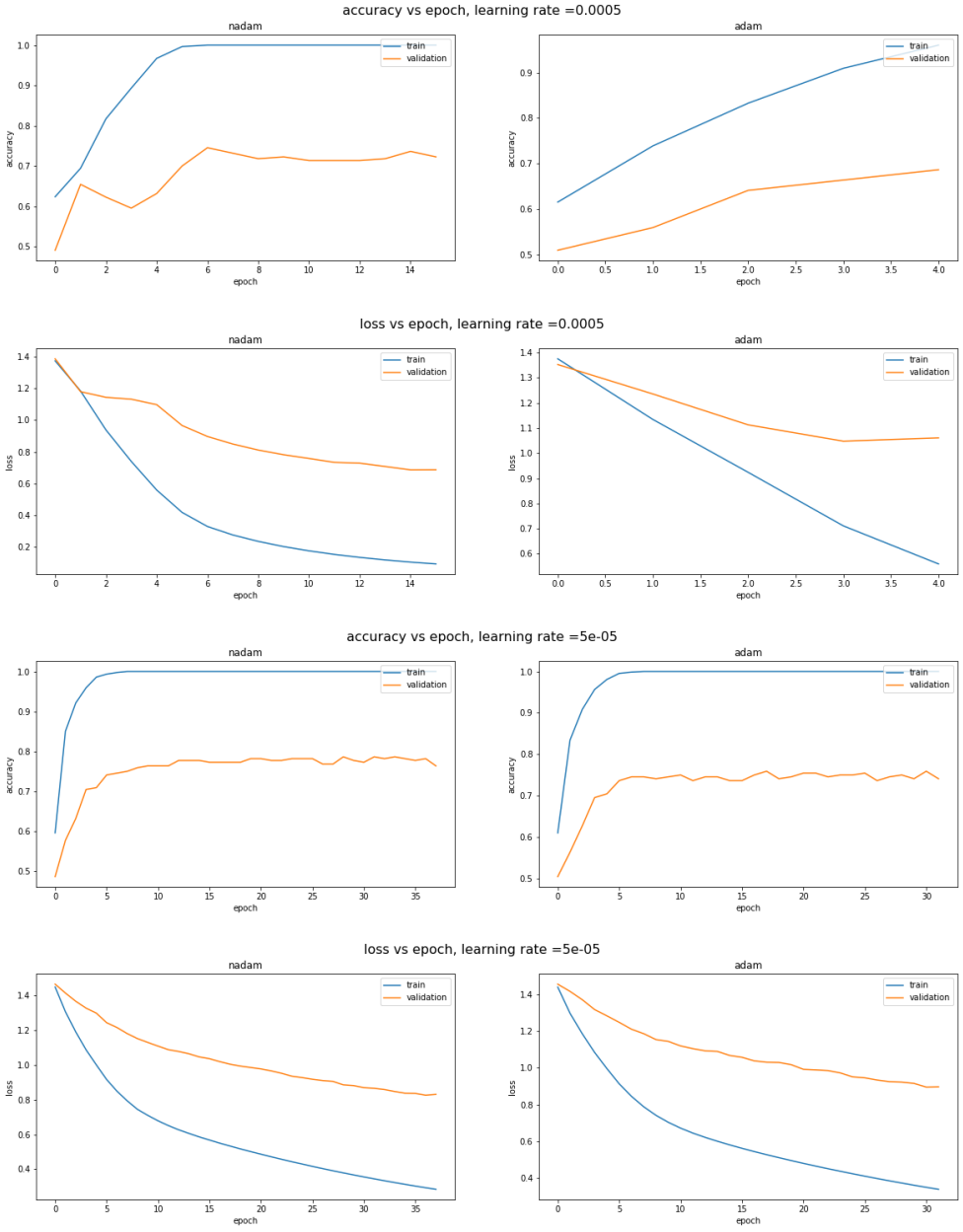
FP: FN: TN: TP:

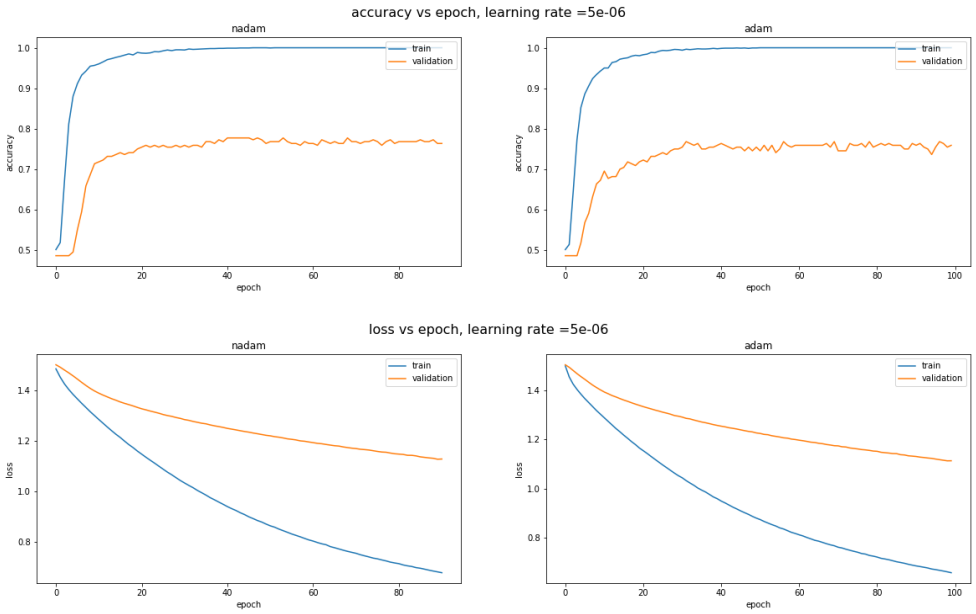
FP example - We think the model was wrong because it identifies two older men, in a suit. In addition, the light background of the first image confused it with the light hair of the person in the second image.
FN example - We think the model was wrong because the man has a very different expression in the first picture, and his face and hair look different.

Second architecture design - with Batch Normalization layers:

hyper parameters	Validation final loss	Validation Accuracy	Test final loss	Test Accuracy	Converge epoch	Training time (seconds)
0.0005 (lr), 'nadam'	**0.685**	**0.722**	**0.695**	**0.732**	**16**	**150.372**
0.0005 (lr), 'adam'	1.060	0.686	1.052	0.648	5	46.277
5e-05 (lr), 'nadam'	0.831	0.763	0.890	0.699	38	349.542
5e-05 (lr), 'adam'	0.896	0.740	0.920	0.726	32	271.348
5e-06 (lr), 'nadam'	1.128	0.763	1.163	0.716	91	834.308
5e-06 (lr), 'adam'	1.112	0.759	1.151	0.717	100	842.024

Performance figures (for each learning rate and optimizer):





The best model with the first architecture is the model with the hyper parameters: 0.0005 (lr), nadam (optimizer) since it had the good accuracy result on the validation set and it converged fast(16 epochs). We will present this model classification matrix and classification examples:

		True class	
		Equal	Not equal
Predicted class	Equal	372	139
	Not equal	128	361

FP: FN: TN: TP:

FP example - We think the model was wrong because it confused one person's hair with another person's hat.
FN example - We think the model was wrong because of the angle at which the person appears in the image.

Conclusion:
As presented, both network architectures yielded similar results, although the second architecture, with Batch Normalization layers, had a small accuracy improvement in the test set. From our experimental process, we found that applying batch normalization can improve the architecture presented in the paper.