Comparing CNNs for Smile Detection

Tabib Wasit Rahman
trahman2@u.rochester.edu

Motivation

Neural networks in general, and particularly Convolutional Neural Networks (CNNs), have very low explainability. They are usually referred to as "black box" models where the inner workings of the model are unknown. This means that it's quite difficult for machine learning experts to understand why the network produced its results. In this project we try to gain insights into the inner working of CNNs by iteratively changing the structure of the model and visualizing the trained kernels. We try to understand how the model behaves by varying the size of the kernels, the number of filters in the convolution layers, and introducing data augmentation to the dataset.

Dataset

The dataset we use is called the CelebA dataset. It contains 202,599 face images of 10,177 unique celebrities. There are 40 binary attributes associated with each face, these include whether the person is male or female, young or old, is smiling or not, has wavy hair, wearing a hat, etc. We focus on one specific attribute: whether the person is smiling or not. The reason for choosing this is partially arbitrary, but we conjectured that smile detection would bring our interesting visual qualities of the trained kernels, such as corner and edge detection filters, and color segmentation filters.

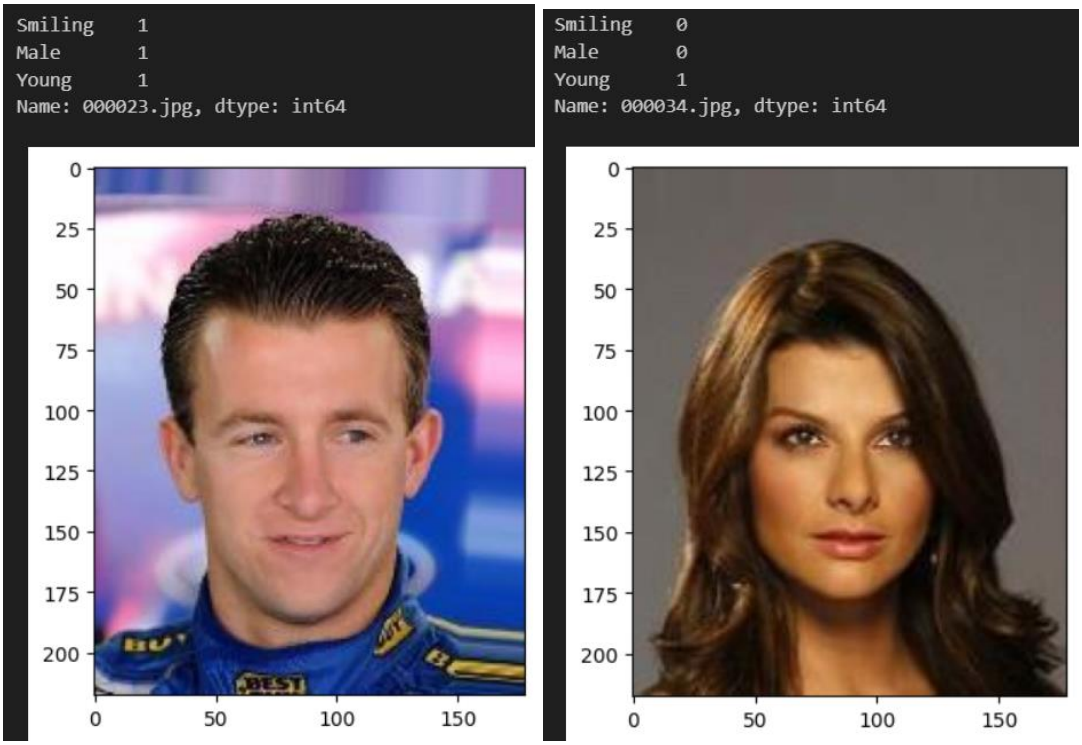

Figure 1: Examples of faces in the dataset

The dataset is partitioned into 162,770 training images, 19,867 validation images, and 19,962 testing images. However, training on all these images would be very resource intensive for our model, so we chose to use a subset of 10,000 training images, 2,000 validation images, and 2,000 testing images.

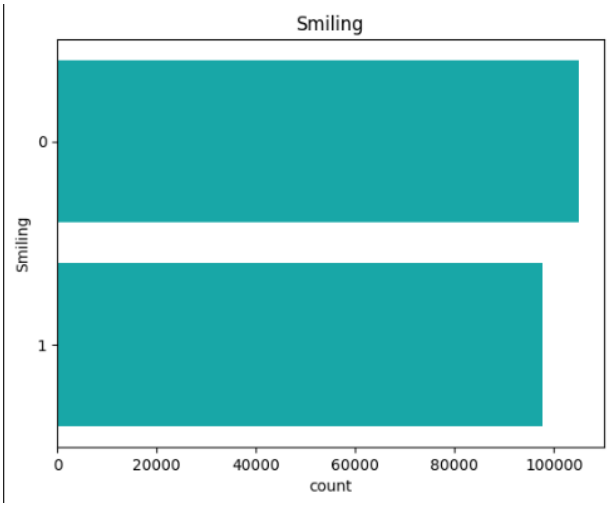

Figure 2: Distribution of smiling attribute in the dataset

The figure above shows that the distribution of the smiling attribute is quite balanced in the dataset, which is a beneficial quality because we know that a balanced dataset provides better results for CNNs.

Method

We create four CNNs to train on our dataset. All the datasets have a common structure: the first two layers are 2D convolution layers with a Rectified Linear Unit (ReLu), with each of the convolution layers being followed by a 2x2 max pooling layer, and then three fully connected layers containing 128 units, 64 units, and finally 1 unit at the output. We apply the sigmoid function at the output to generate a probability, which is the probability of the face in the image to be smiling. We train the model for 10 epochs with early stopping, which means that we stop training the model when the accuracy on the validation set starts to decrease. We perform backpropagation using the Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.001 and momentum of 0.9. We use binary cross entropy loss for the optimization.

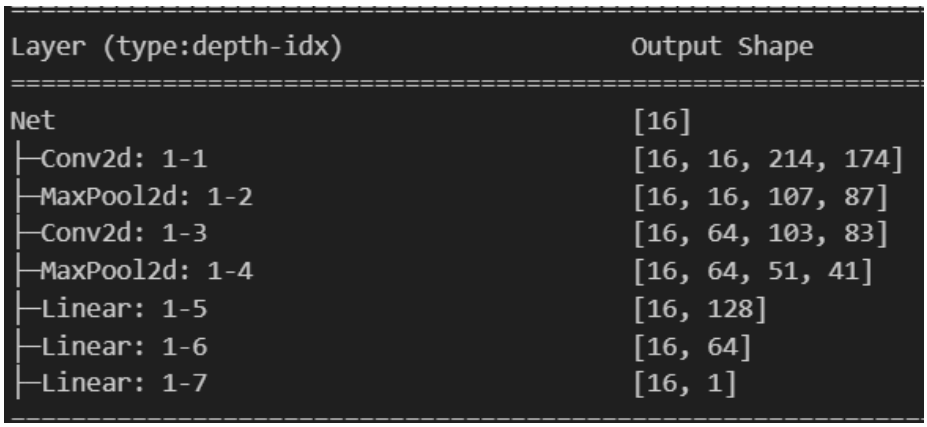

Figure 3: Model 1 with a 5x5 kernel

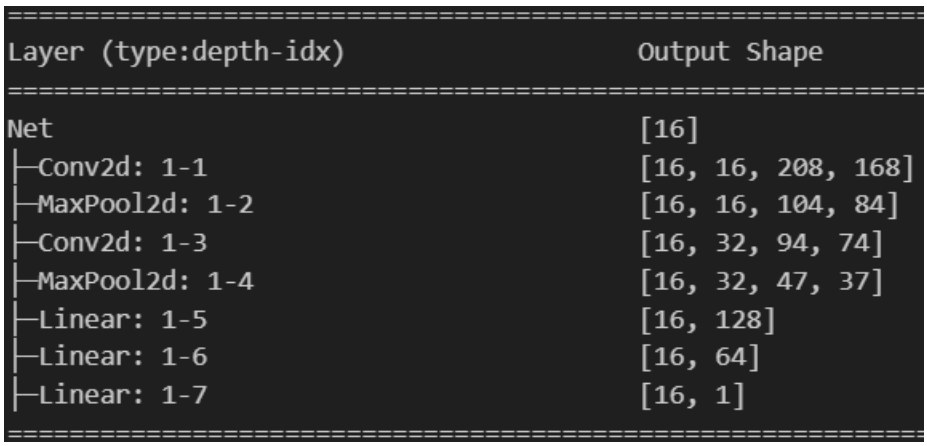

Figure 4: Model 2 with a 11x11 kernel

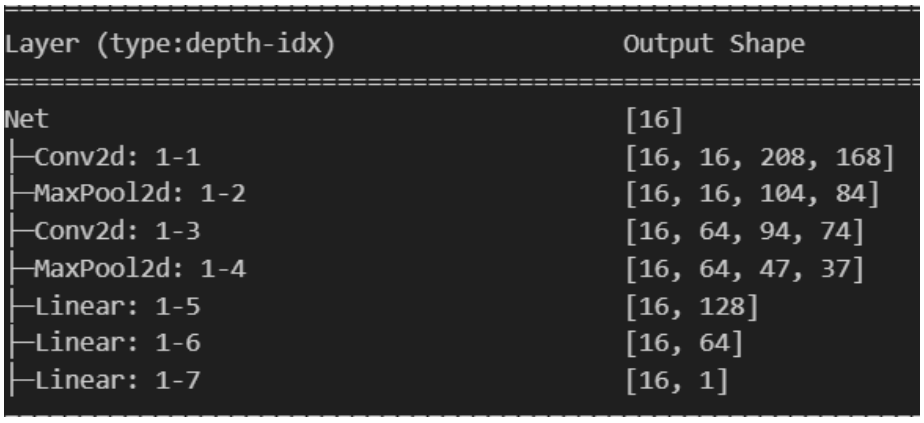

Figure 5: Model 3 with 11x11 kernel but with twice the number of filters in the second convolution layer, and model 4 with 11x11 kernel using data augmentation

Figure 3 to 5 describe the four models that were used in this project. For model 3 the dataset was normalized with each of the RGB channels having a mean of 0.5 and a standard deviation of 0.5. For model 4 the data was augmented using a random affine transform with a rotation of 30 degrees, 20% translation, 70% scale and 20% horizontal shear. A random horizontal flip transform was also used with the affine transform.

Results

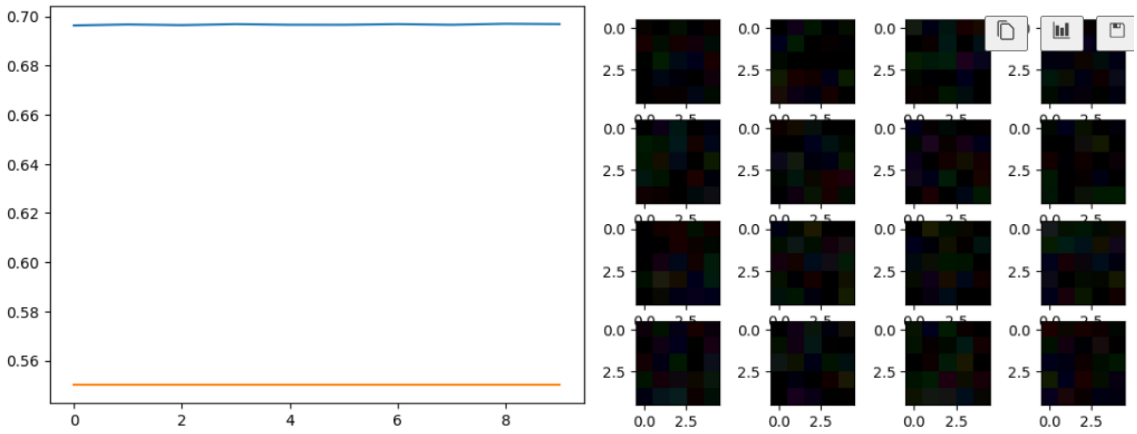

Figure 6: The results of model 1

Model 1 performed poorly during training. The training loss did not decrease and stayed at around 0.7 and the accuracy on the validation set was around 0.55.

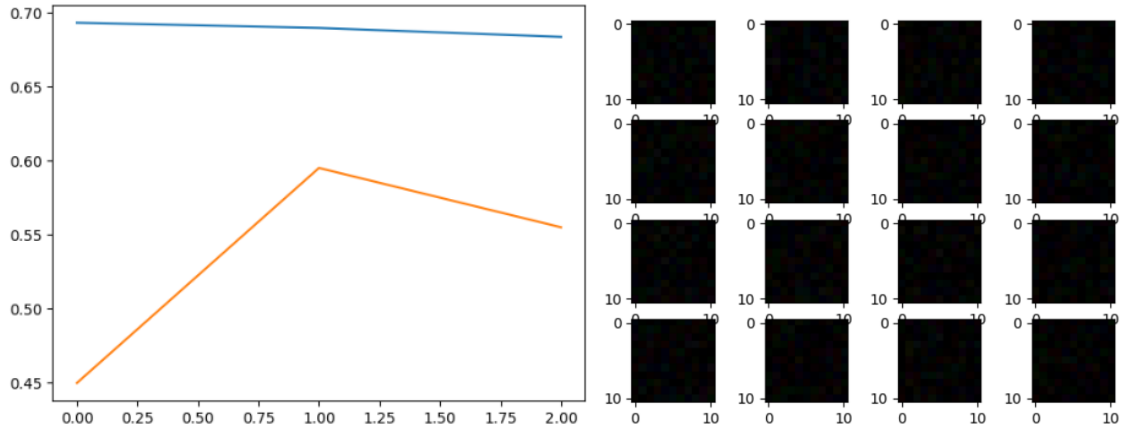

Figure 7: The results of model 2

Model 2 performed slightly differently. We see the training loss decreasing slightly, and the validation accuracy increasing. However, due to early stopping the training stops at the third epoch.

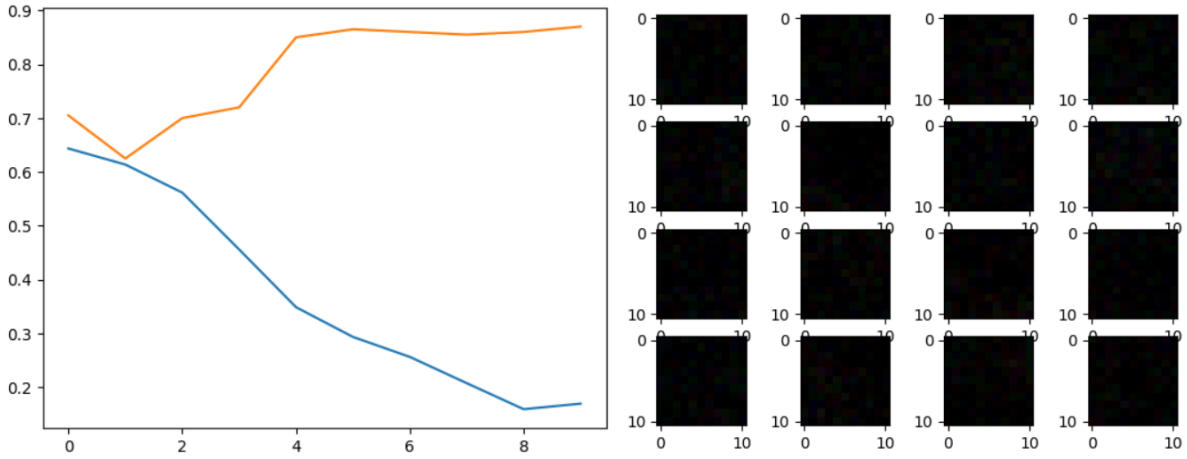

Figure 8: The results of model 2 without early stopping

When we remove early stopping, we see that within 10 epochs the model can achieve a higher validation accuracy and a lower training loss. The peak validation accuracy is around 0.87.

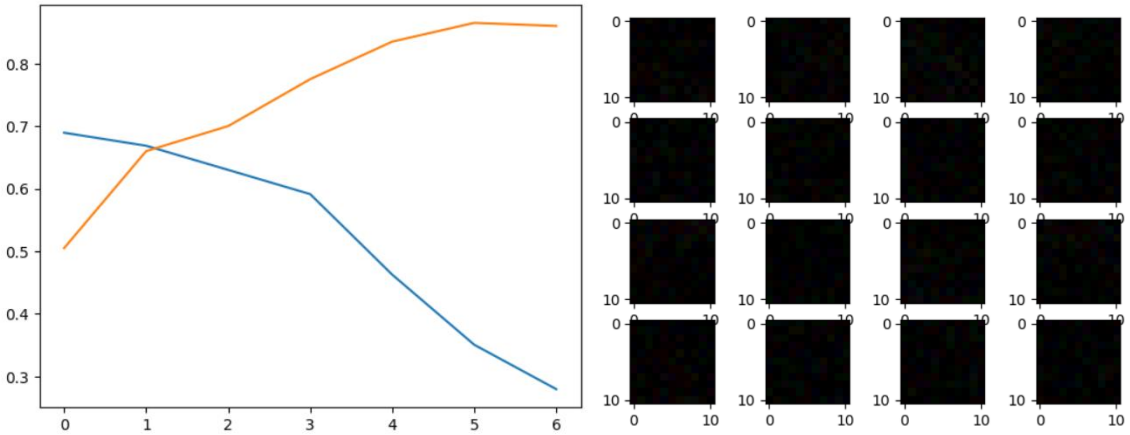

Figure 9: The results of model 3

We can see that with model 3, the data normalization makes the training smoother, and the model is also able to achieve a peak validation accuracy of around 0.86.

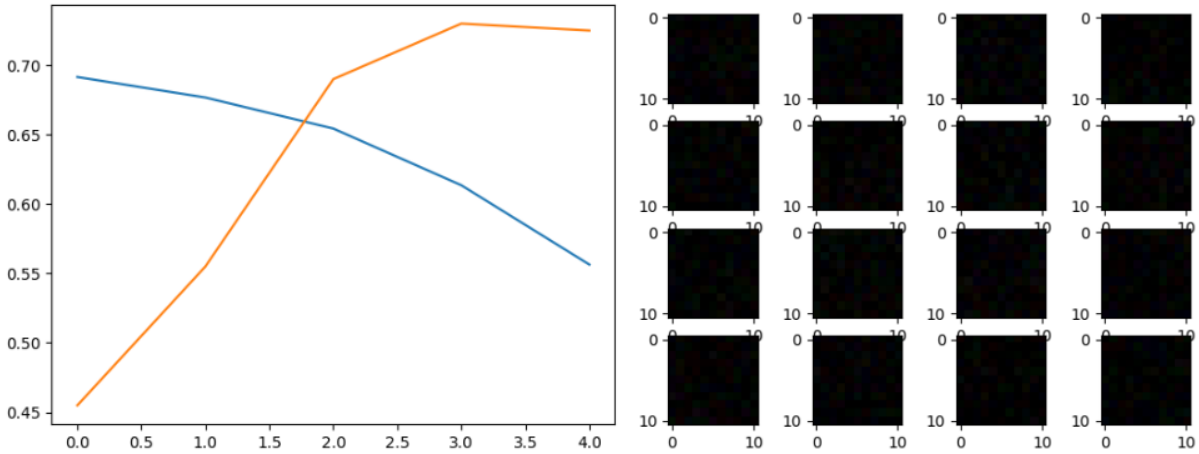

Figure 10: The results of model 4

Model 4 performs slightly worse than model 3, reaching a validation accuracy of 0.8 which suggests that the data augmentation was not favorable to the model training.

Conclusions

In conclusion, we can see that kernel size greatly impacts the model accuracy; using a larger kernel size of 11x11 instead of 5x5 allowed the model to be successfully trained. Normalizing the data had a positive effect on the model, it allowed the model to be trained more smoothly. However, we can agree that varying the number of filters in the second convolution layer and augmenting the data with an affine transform did not have a significant effect on the model accuracy. We do note that increasing the kernel size and the number of filters greatly increases the training time of the model, which means that this creates a bottleneck in effectively visualizing the kernels and increasing the model size due to the restriction of the GPU resources.

References

https://www.kaggle.com/datasets/jessicali9530/celeba-dataset