Title of the project: Weather Classification from Videos/Images using Convolution Neural Networks

Team: Bhuwan Sapkota, bhuwan.sapkota@ucdenver.edu
Samrid KC, samrid.kc@ucdenver.edu

1.Introduction

In this project we analyzed video clips and estimated the weather based on the frames derived from those videos, using Convolution Neural Networks. The project is interesting because it can be used in AI projects such as,

- Driver assistance systems to make informed decisions based on the information that the model provides. These systems can take in real time video and be fed the frames to analyze the weather.
- This weather estimation model can also help applications that colorize old black and white pictures. If we can estimate the weather in a black and white picture, this can help the colorization to be more accurate.
- The model can also be helpful in categorizing untagged images in large databases. For example, it can used to tag different landscapes from images of landscapes from websites, databases etc.
- The model can be used to categorize images according to seasons in cloud-based applications like Google Photos.

The Convolution Neural Network that we implemented in this project is mostly successful in correctly predicting the weather in a video clip.

2. Problem Definition and Algorithm

2.1 Task Definition

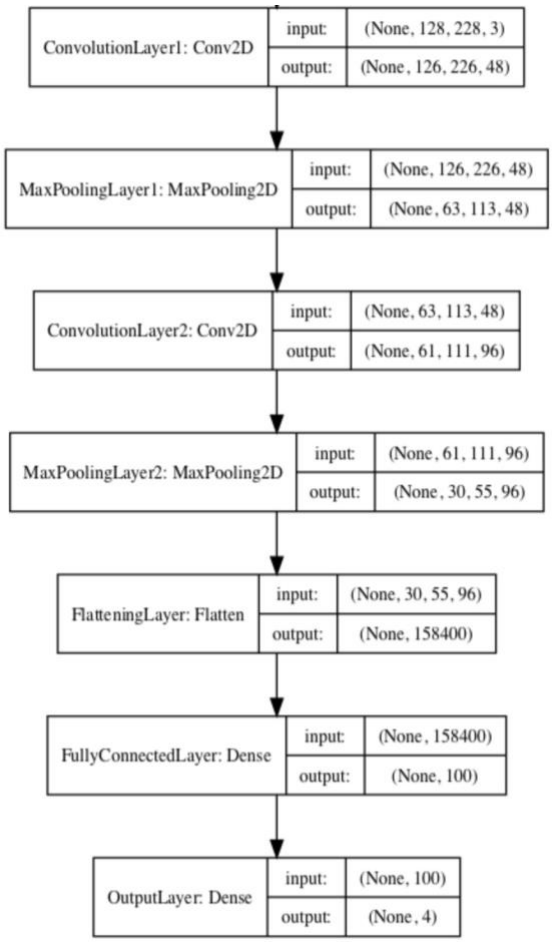
The JAAD (Joint Attention for Autonomous Driving) Dataset contains 346-high resolution video clips extracted from driving videos. All of these videos are tagged by various attributes such as time of day, weather, location etc. We utilized the weather tags (clear, cloudy, rain, snow) to train our model to accurately predict the weather from video clips.

2.2 Algorithm Definition

From the behavioral data that was provided with the dataset, we extracted the weather tags for each video from tsv files and assigned the tags to the videos. Most of the videos were 30fps, and there was very little change between these frames, so we only extracted 1 frame per 30 frames from each video. These frames were all RGB (3 channels). The frames were initially of the resolution 1080*1920. We reshaped these images into 128*228. The data was preprocessed and split into training and validation with a 70-30 split. Before being input into the model, the

training and validation data was also zero centered. The project utilizes Keras to build and train the model. The model architecture is as follows,

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
ConvolutionLayer1 (Conv2D)   (None, 126, 226, 48)      1344
_________________________________________________________________
MaxPoolingLayer1 (MaxPooling (None, 63, 113, 48)       0
_________________________________________________________________
ConvolutionLayer2 (Conv2D)   (None, 61, 111, 96)       41568
_________________________________________________________________
MaxPoolingLayer2 (MaxPooling (None, 30, 55, 96)        0
_________________________________________________________________
FlatteningLayer (Flatten)    (None, 158400)            0
_________________________________________________________________
FullyConnectedLayer (Dense)  (None, 100)               15840100
_________________________________________________________________
OutputLayer (Dense)          (None, 4)                 404
=================================================================
Total params: 15,883,416
Trainable params: 15,883,416
Non-trainable params: 0
_________________________________________________________________
None
```

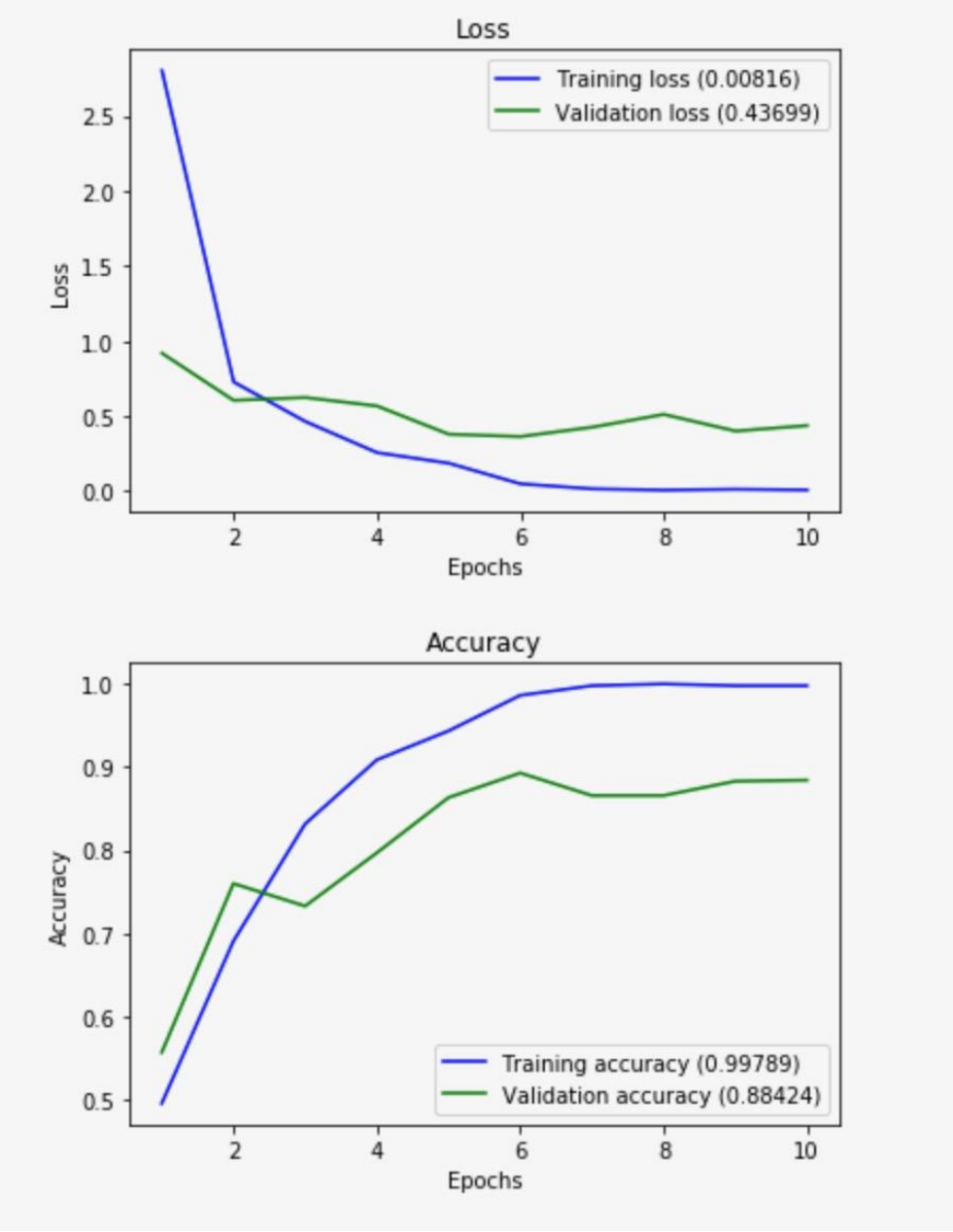


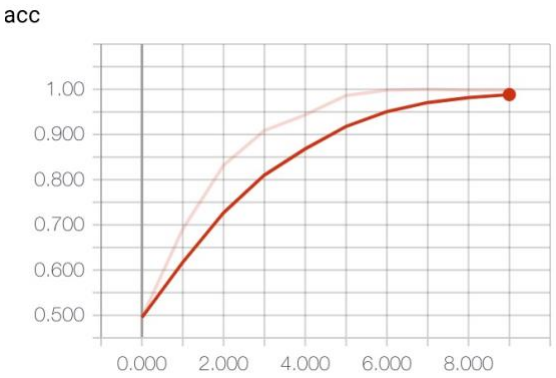
3. Experimental Evaluation

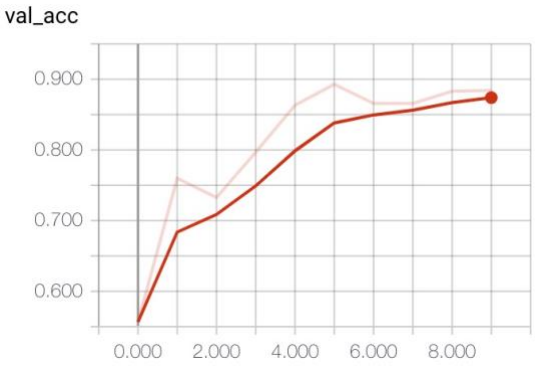
3.1 Methodology

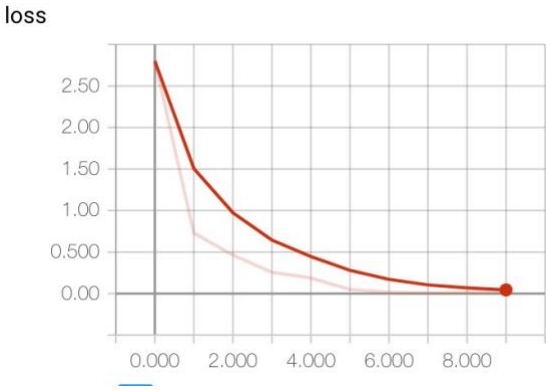
To evaluate the results of the network, we will evaluate the model by examining the plots for accuracy and loss metrics. Epoch loss plot for training and validation will also be analyzed. We also calculated the confusion matrix for our model to visualize the performance of the model.

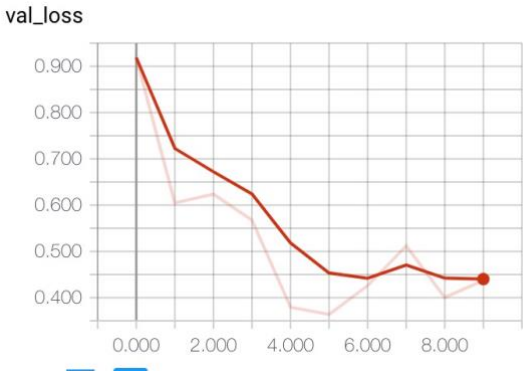
3.2 Results

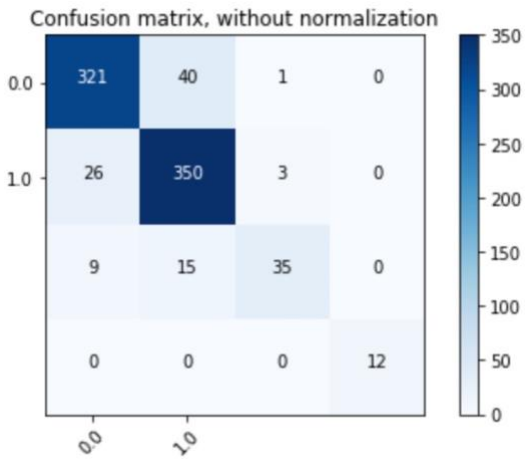


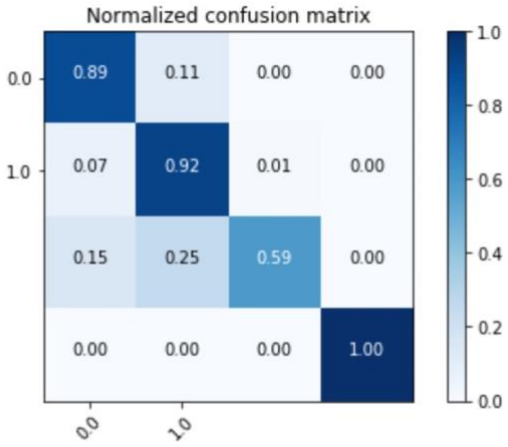












3.3 Discussion

As the above evaluation methods show the weather prediction model that was implemented in this project is accurate up to around 90% of the time. The model sometimes predicts videos tagged as rainy to be cloudy as the image features for both these categories are similar. The dataset also only contained 3 short video clips of snowy weather, therefore we do not feel confident that it will accurately predict snowy weather in comparison to the other weather tags.

4. Related Work

We did not compare the performance of the model implemented in our project with any other similar works.

5. Future Work

The dataset was of a smaller size than is used for most complex deep learning projects. If we were to work on bettering our model for weather prediction in the future, we would collect more data and train the model accordingly. The smaller size of data also led to overfitting when the model was being trained, therefore, we were constrained to keep the number of epochs to a smaller number. With a larger dataset, we would be able to train the model for a substantially more number of epochs. This would lead to the model being trained to make even better predictions.

6. Conclusion

We were successful in analyzing video clips and estimating the weather based on the frames of these video clips. The model that was implemented in this project utilizes Convolution Neural Networks. The network predicts weather up to an accuracy of around 0.9, and we think that if it is trained with a larger dataset it would make predictions with an even higher accuracy.

Bibliography

[1] https://www.cs.ccu.edu.tw/~wtchu/papers/2016BigMM-chu.pdf(Last accessed 11-25-2018)

[2] https://ieeexplore.ieee.org/document/7351424(Last accessed 11-25-2018)

[3] https://arxiv.org/pdf/1808.00588.pdf(Last accessed 11-28-2018)