Comparative Analysis of Deepfake Detection Models

Maximilian Huber, Kevin Tang, Brevinh Pham
Khoury College of Computer Science
Northeastern University

***Abstract*— In this project we aim to conduct a comprehensive comparative analysis of deepfake video detection models, leveraging the open-source Deepstar toolkit. This toolkit provides training data and two example neural networks for deepfake detection. Our goal is to train the two networks and compare their effectiveness, and then use our insights to design our own model and compare it to the original two. We will speculate about the impact a given model's structure has on the given model's performance. Our aim is to improve our model to be as accurate as possible, and ideally at least as accurate as the two examples.**

I. INTRODUCTION

A deepfake is an image or video of a person in which their likeness has been digitally altered so that they appear to be someone else. Deepfakes can be used as malicious or destructive tools by bad actors, as they are often indistinguishable from real pictures and videos to the human eye. Deepfakes have already been used to commit crimes of misinformation and identity fraud, and they are only getting more dangerous as the technology improves. As deepfakes become harder and harder to discern from reality, it is critical that the models which we use to detect them evolve as well.

Deepstar is an open-source toolkit created by ZeroFox for the specific purpose of helping researchers to create better deepfake detection models. It offers a dataset of both real and deepfaked videos (packaged as a set of YouTube URLs), and it also provides two example detection models for researchers to use as a baseline. The two example models are called Mouthnet and Mesonet.

Our goal for this project is to train and test Mouthnet and Mesonet on the video dataset provided by Deepstar, and then to use our observations to build a new model which performs even better.

II. DEEPSTAR MODELS

Both Mouthnet and Mesonet are convolutional neural networks (CNNs) which classify individual frames of a video independently. A CNN is the most prevalent approach to image classification, as the convolutional layers are very effective at transforming image data into a rich feature representation which a standard feed-forward neural network can more easily classify.

CNN's extract information from images by passing a small filter over the image, and using the filter to enhance features of the image which match the filter. This is known as convolution, and is carried out by a convolutional layer. This process leads to a higher contrast image, where any relevant features (usually edges) are even more pronounced and sharp. Because the image's features are more pronounced, the image can be scaled down without losing these important features. This is usually achieved with a max pooling layer, which takes the maximum value of a group of pixels in the original image, and makes that the corresponding pixel value in the new lower resolution image. This process of passing an image through convolution layers and max pooling layers is usually repeated multiple times, until eventually the image is a much more compact feature representation of its original self. Then the image can be flattened into a vector and passed through densely connected layers to create a classification.

A. *Mesonet*

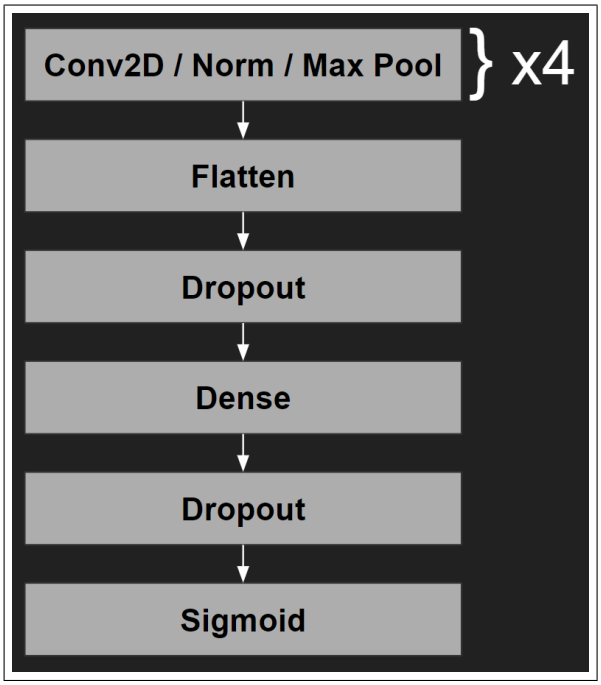

Fig. 1. Mesonet Architecture

The first of the two provided Deepstar models is called Mesonet [1]. Mesonet is a CNN with four repeating sets of convolutional layers, normalization layers, and max pooling layers which reduce an image into a rich feature

representation. The image is then flattened into a vector, and passed through a dropout layer (which randomly masks parts of the vector passed to it). Then the vector is passed through a dense layer (a dense layer is the backbone of most neural networks, it is a layer that takes in data, weighs it, adds a bias, applies an activation function, and produces an output that's passed to the next layer for further processing), then another dropout layer, and finally a single sigmoid function. This sigmoid function ensures that the final output of the model falls in the range of 0 and 1, which allows us to map the model output to our binary classification task (if a video is fake the model is expected to output 0 and if the video is real the model is expected to output 1).

B. *Mouthnet*

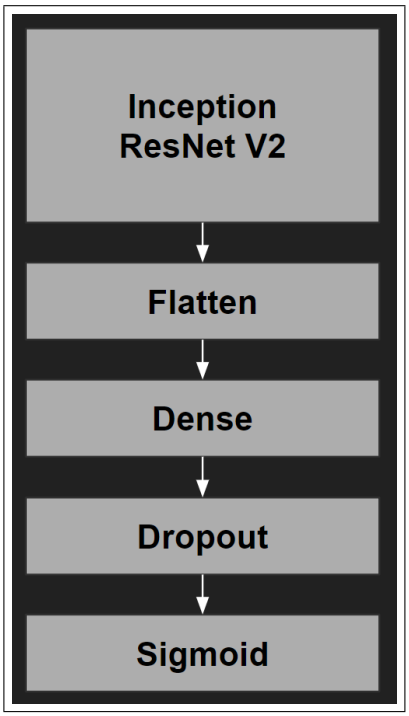

Fig. 2. Mouthnet Architecture

Mouthnet, the other model provided by the Deepstar toolkit, is also a CNN like Mesonet, but with a few key differences. The most stark difference is that the convolutional tail of Mouthnet is actually a pre-existing convolutional feature extraction architecture called *Inception ResNet V2*. Inception ResNet V2 differs from the set of convolutional layers used by Mesonet mainly in that it features *residual connections*, which allow for inputs to "pass through" certain layers and recombine with their own feature representations in later layers [2]. The rest of Mouthnet is almost the same as Mesonet, with the feature representation produced by Inception ResNet V2 being flattened into a vector, passed through a dense layer, passed through a dropout layer, and then fed to a sigmoid function for the final binary classification output.

Mouthnet, having roughly 61,000,000 tunable parameters, is considerably larger than Mesonet, which only has around 28,000. About 54,000,000 of Mouthnet's parameters are just from the Inception ResNet V2 model. This difference in scale between Mesonet and Mouthnet makes a fair comparison of their utility difficult.

C. *Initial Performance of Deepstar Models*

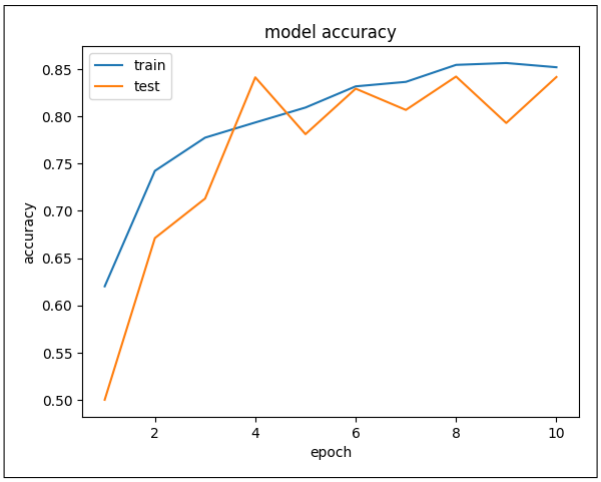

Fig. 3. Initial Mesonet Accuracy

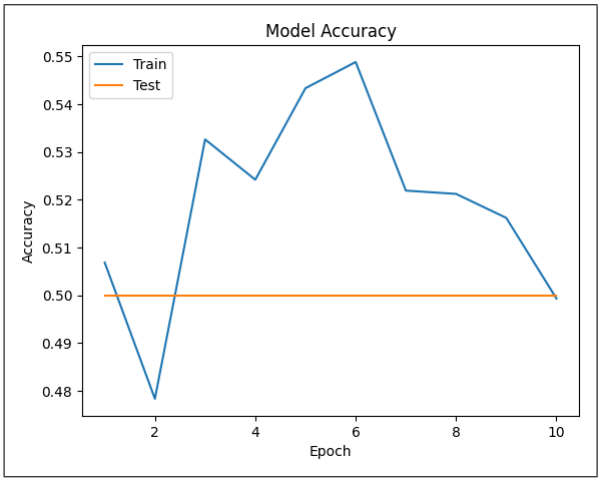

Fig. 4. Initial Mouthnet Accuracy

As seen in Fig. 3 and Fig. 4, Mesonet exhibited a much more consistent initial performance than Mouthnet. The Mesonet accuracy on both the training and test set steadily improved over time, with a modest best test accuracy of 84.22%. Mouthnet on the other hand did not improve at all, and the suspicious constant 50% test accuracy leads us to believe that the model was most likely classifying every sample in the test set as a single class. Given that Mouthnet has a different optimizer (Stochastic Gradient Descent) than the one used by Mesonet (Adam), and the fact that Mouthnet is orders of magnitude larger (in terms of learnable parameters), it is possible that Mouthnet simply needs way more training time than it

was given, and *could* eventually start producing accurate classifications. Mouthnet took 1210 seconds per epoch on average, whereas Mesonet took only 568 seconds per epoch on average. Mesonet is the clear winner.

III. POTENTIAL ISSUES

Following our initial training and testing of the Deepstar models, we identified two main issues which we sought to remedy to improve our own model's accuracy.

A. *Frame Sampling Issue*

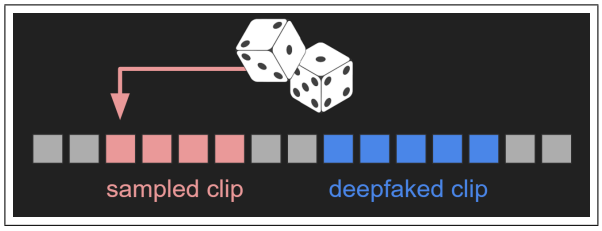

Fig. 5. Random Clip Sampling

One problem we identified was related to how the data was being sampled from the videos. Because the Deepstar models are essentially image classifiers, they have to be fed individual frames from the videos. The current method of collecting frames from a video involved randomly selecting a clip from the video with a predefined length, and then using the true label of the video ("fake" or "real") as the true label for all the frames in the clip.

The main issue with this is that the deepfaked segment of a video is very rarely the length of the whole video. So when the frame sampler randomly selects a clip from a video, there is a chance that some or even all of the frames sampled do not contain any deepfaking, even when the video is labeled as deepfaked. This leads to the possibility of the models being trained largely on mislabeled frames, which will have a problematic effect on its performance.

B. *Temporal Context*

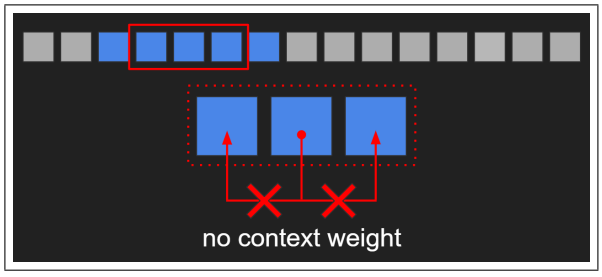

Fig. 6. Temporal Context

The problem with poor frame sampling is further compounded by the second and more obvious issue we identified: the Deepstar models have no temporal context. Because both Deepstar models are image classifiers, each frame of a video must be classified independently of all other frames, with the model being unable to use the information of the preceding or following frame to help make its decision. Since we are attempting a video classification task, this is a big problem. Using our current frame sampler, it is possible that some of the frames in a video will be deepfaked and some will not, but because the models are classifying each frame independently, it will not be able to identify that the non-deepfaked frames belong to a deepfaked video, even if it correctly identified the actually deepfaked frames in the same video.

IV. PROPOSED SOLUTIONS

A. *Multi-Clip Sampler*

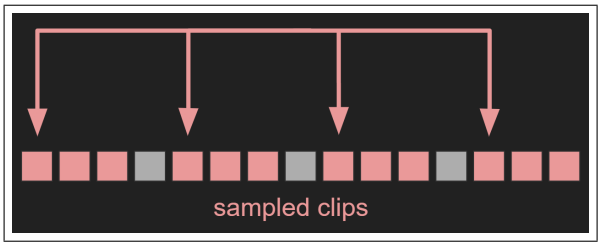

Fig. 7. Multi-Clip Sampling

To remedy the problems caused by random frame sampling, we created a new data generator. This generator splits the video into multiple, evenly-spaced clips throughout the length of the video. This way, no matter where the deepfaked clip is, there is a better chance that at least some deepfaked frames will be included.

B. *RNN*

Our first attempt at solving the lack of temporal context was to augment Mesonet (the better performing Deepstar model) with a new temporally aware architecture. This was achieved by implementing a kind of architecture known as an RNN.

An RNN, or Recurrent Neural Network, is a type of neural network that excels at understanding and making predictions about sequences of data. Unlike the layers of a standard feed-forward network, the blocks of an RNN are capable of sharing information laterally, which allows the previous samples of a sequence to have an effect on the output for the current sample. This gives us the temporal context we need, as the decisions generated for previous frames can be used to influence the decision made for the current ones. Because an RNN can process sequences, we can now feed it all the frames sampled from a single video and generate a single classification for that video, which is exactly aligned with our goal.

Our specific approach for turning Mesonet into a temporally-informed model can be seen in Fig. 8. All the sampled frames of a given video are fed into a time-distributed, pretrained Mesonet model. This results in a set of confidence values, where each value indicates Mesonet's confidence that the corresponding frame is real and not deepfaked. This sequence of confidence values is then passed to an LSTM, which is our model's RNN component.

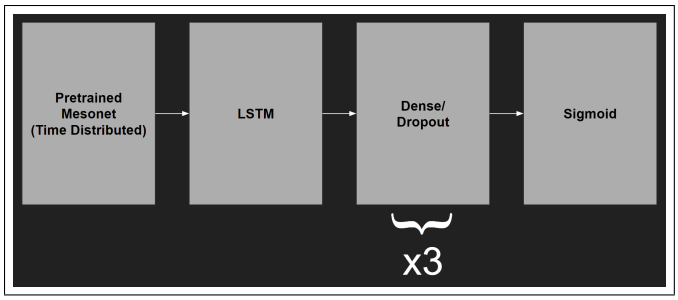

Fig. 8. RNN Architecture

An LSTM, or Long Short Term Memory, is a kind of intelligent memory unit for a neural network. It can remember, update, and forget previous information in the sequence, both in long and short term. By using this component to process our sequence of confidence values, we can generate a feature representation of our video which takes into account all of the frames in our video, which gives us the temporal inference we were missing before. This representation can then be processed similarly to how we processed image feature representations in Mesonet and Mouthnet. The vector is passed through an alternating set of three dense and three dropout layers, before being passed to the sigmoid function for a binary classification output.

C. *Optical Flow*

Another solution to the lack of temporal context that we experimented with was the use of feature extraction via Gunnar-Farneback optical flow.

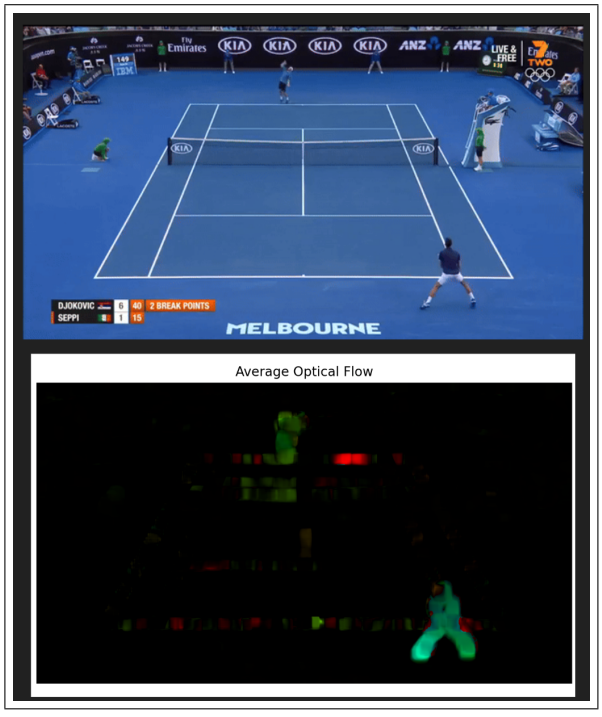

Fig. 9. Optical Flow Example

Gunnar-Farneback optical flow is an algorithm that estimates the motion between two images. It does this by approximating the image brightness patterns with quadratic polynomials and calculating the displacement fields that align these patterns from one image to the next[3]. This method involves smoothing the images, expanding them into polynomial terms, and solving a set of linear equations to find pixel displacements (i.e. how far a pixel is displaced from one frame to the next). This allows the algorithm to generate images which capture the motion between frames, which can then be used as features for the video classification task. The caveat of this is that the model is effectively "colorblind" because the algorithm replaces the original color information of the frames.

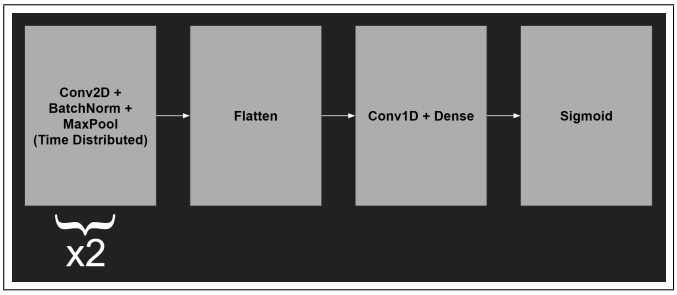

Fig. 10. Optical Flow Model Architecture

To establish a model incorporating optical flow feature extraction, we first generated optical flow videos from each entry in our original dataset provided by Deepstar. Then we developed a new model better tailored to leveraging optical flow, diverging from the traditional RNN framework. The modifications included adopting a custom CNN for feature extraction (in place of the Mesonet model), and utilizing a 1-D convolutional layer instead of an LSTM to process the sequential features. This adjustment was predicated on the hypothesis that the 1-D convolutional layer would more effectively discern short-term patterns, which are crucial for detecting deepfakes. To continue, we utilized the Multi-Clip Sampling technique to enhance the model's efficacy in identifying these discrepancies.

V. EXPERIMENTAL RESULTS

A. *Model Parameters*

1) **Random Sampler:**
   a) **Frames per Video:** 100
2) **Multi-Clip Sampler:**
   a) **Frames per Sample:** 10
   b) **Samples per Video:** 10

Model	Epochs	Batch Size	Optimizer	Learning Rate
Mesonet	10	10	Adam	0.001
Mouthnet	10	10	SGD	0.1
RNN	10	2	Adam	0.001
Optical Flow	10	16	Adam	0.001

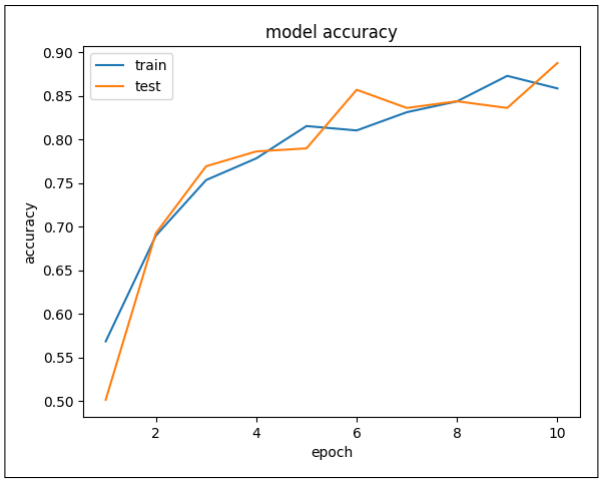

Fig. 11. Mesonet Accuracy with Multi-Clip Sampling

B. *Multi-Clip Sampler*

Comparing the performance of the Deepstar models using the new data generator to the original, we see a small improvement in the results. The highest test accuracy for Mesonet while training improved from 84.22% to 88.74% using the new sampling method, while the Mouthnet test accuracy remained at a constant 50%. This small improvement for Mesonet was most likely caused by a select few samples in the dataset in which frames had been "mislabeled" due to the randomness of the previous sampling method. While it was not enough to trigger the Mouthnet to start learning properly, the slight improvement to Mesonet was encouraging.

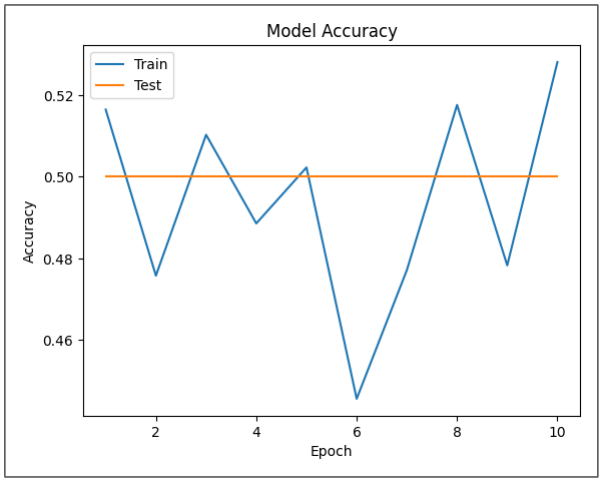

Fig. 12. Mouthnet Accuracy with Multi-Clip Sampling

C. *RNN*

As seen in Fig. 13, the RNN gradually improved as training time went on, and outperformed both Mesonet and Mouthnet in terms of test accuracy while training with both the old random frame sampler and the new multi-clip sampler. The highest test accuracy it reached while training with the old sampling method was 91.67%, but even that was not as impressive as the highest test accuracy from training with the new multi-clip sampler. In the final epoch of training, the RNN with the multi-clip sampler achieved 100% testing accuracy. In theory, this should mean that the RNN is capable of correctly classifying videos it has not seen before 100% of the time. While this is obviously not true, it is still very encouraging, and definitively shows that a temporal dimension is invaluable when classifying sequential input like video. The RNN has clearly mastered the Deepstar dataset, and is ready to be trained and tested on an even larger and more diverse dataset, to further generalize the model.

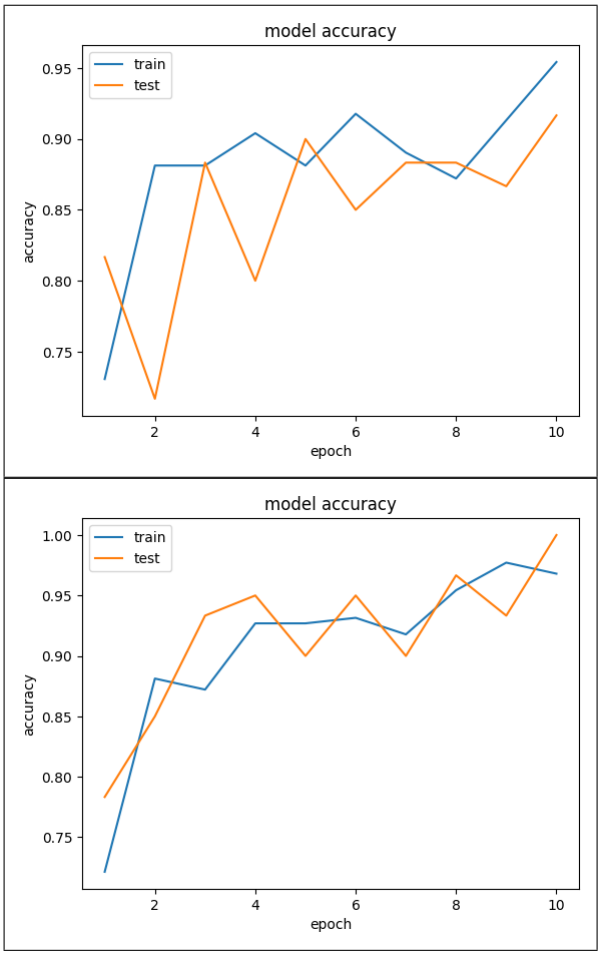

Fig. 13. RNN Accuracy with Random Sampling (top) and Multi-Clip Sampling (bottom)

D. *Optical Flow Model*

As seen in Fig. 14, the optical flow model did improve in accuracy over time, but ultimately resulted in a peak test accuracy worse than that of the RNN model. The highest test accuracy achieved during training was 73%. Additionally, the test accuracy does not correlate closely with the train accuracy, which most likely indicates that this model over-fitted to the training data.

The disappointing performance could be the result of several factors, the first of these being that we did not

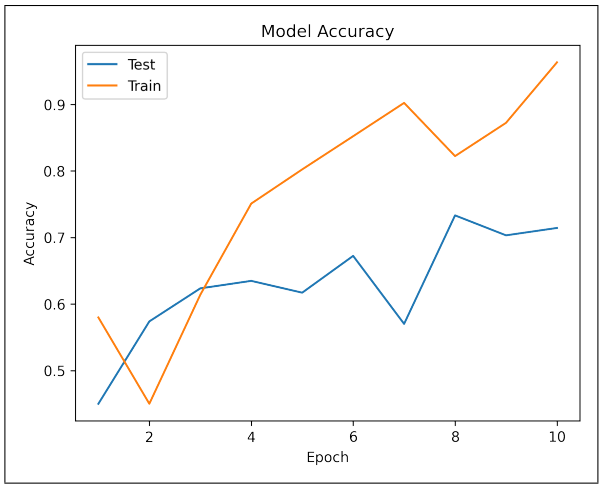

Fig. 14. Optical Flow Model Accuracy with Multi-Clip Sampling

implement a two-stream model[4]. Typically, when using optical flow in computer vision classification problems, one would use a two stream model which combines a temporal model using optical flow and a spatial model using image classification. However, due to the high computational cost of converting videos to optical flow, we decided to just test a single stream model first. Because each video must first be converted to an optical flow representation, it takes an incredible amount of compute to preprocess to data. With optimal parallelization on an A100 GPU, six minute videos could take upwards of 20 minutes to convert into an optical flow representation.

Another factor that may be reducing the optical flow model's performance is a result of the Gunnar-Farneback optical flow algorithm itself. Since it uses a polynomial model to estimate motion, the algorithm tends to be worse at interpreting small, discrete motions. This means that Gunnar-Farneback optical flow has trouble encoding subtle facial movements, which is crucial for detecting deepfakes.

In conclusion, although a two-stream optical flow model *might* surpass the performance of the RNN, several considerations dissuade us from pursuing this approach: the substantial computational resources required and the potential limitations of optical flow in addressing this specific problem present significant drawbacks. These factors collectively suggest that the two-stream model may not provide a feasible or efficient solution under the current constraints.

VI. DISCUSSION

A. *Results Analysis*

The results of our experiment suggest that two of our proposed solutions (Multi-Clip Sampling and RNN) had a positive impact on overall task performance. The RNN outperformed the two Deepstar models, and the new frame sampling method lead to slight improvements for both Mesonet and RNN (Mouthnet remained a lost cause).

Optical flow showed intriguing potential, but ultimately required too much computational overhead to make it effective. It also seemed to struggle with detecting facial movements, which makes it not a great fit for this task.

B. *Challenges*

This project was not without challenges. Getting the Deepstar models to train or classify at all was a bit of an arduous process, in no small part due to the lack of any substantial documentation for the Deepstar Toolkit. It is possible that our reimplementation of Mouthnet is part of what caused its terrible performance.

C. *Future Work*

Since the RNN reached a perfect validation accuracy, any further testing should be conducted on a larger, more diverse dataset. The new frame sampling method appears to be better, but it is still not perfect, as frames from different parts of the video appear to be next to each other (as a result of stitching together multiple clips).

VII. GITHUB REPOSITORY

`https://github.com/MaxHuber888/DeepSquid`

VIII. INDIVIDUAL CONTRIBUTIONS

A. *Maximilian Huber*

- Set up Mesonet, Mouthnet, and Frame Sampler
- Designed and implemented the Multi-Clip Sampler
- Designed and implemented the RNN
- Drafted Report

B. *Kevin Tang*

- Organized and oversaw all Mesonet, Mouthnet, and RNN training
- Generated all Mesonet, Mouthnet, RNN plots/values
- Refactored/Organized repository

C. *Brevinh Pham*

- Organized and oversaw all Optical Flow Training
- Designed Optical Flow Converter and Network
- Generated all Optical Flow Parts and Values
- Drafted Report

REFERENCES

[1] Afchar, D., Ecole des Ponts Paristech, JFLI, CNRS, UMI 3527, Japan, LIGM, UMR 8049, UPEM, France, Nozick, V., National Institute of Informatics, Yamagishi, J., & Echizen, I. (n.d.). MesoNet: a Compact Facial Video Forgery Detection Network. Cornell University. https://arxiv.org/pdf/1809.00888.pdf

[2] Szegedy, C., Google Inc., Ioffe, S., Google Inc., Vanhoucke, V., Google Inc., Alemi, A., & Google Inc. (2016). Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning [arXiv:1602.07261v2 [cs.CV] 23 Aug 2016]. Cornell University. https://arxiv.org/pdf/1602.07261v2.pdf

[3] Fast and accurate motion estimation using orientation tensors and parametric motion models. (2000). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/905291

[4] Khan, S., Hassan, A., Hussain, F., Perwaiz, A., Riaz, F., Alsabaan, M., & Abdul, W. (2023). Enhanced spatial stream of Two-Stream Network using optical flow for human action recognition. Applied Sciences, 13(14), 8003. https://doi.org/10.3390/app13148003