# Final Project Proposal

## Definitions

**Team name:** ***Maximum Likelihoodlums***

**Team members:** ***Cheng-You Lu, Kelly Patel, Ji Won Chung***
*Note:* Once one person uploads the report to Gradescope, please add all other team members to the submission within the Gradescope interface (top right on your submission).

If you need to find team members, please use the 'Search for Teammates!' top-level post on Piazza—pitch an idea!

## Project

Please write a one-two page document including:

- What are the skills of the team members? Conduct a skill assessment!
- What is your project idea?
- What data will you use?
- What software/hardware will you use?
- Who will do what?
- How will you know whether you have made progress? What will you measure?
- What technical problems do you foresee or have?
-
- What is the socio-historical context that this project lives in? (2-3 sentences)
- Who are the stakeholders for this project? (3-4 sentences)
- What are the benefits of a technology such as this? (2-3 sentences)
- How might a bad actor misuse this technology and who would it harm? (2-3 sentences)
-
- Is there anything that we can do to help? E.G., resources, equipment.

Feel free to use these as paragraph headings, and also please include any media, references, etc.

# 1 Proposal

## 1.1 Team Members' Skills

All team members are graduate students with a bachelor's in Computer Science, but this is their first time taking computer vision. All three members have experience in python.

Cheng-You possesses a little deep learning experience (e.g., Tensorflow and Pytorch) in weakly-supervised semantic segmentation and video rescaling network with the invertible neural network. Kelly does not have much experience with deep learning or neural networks, but has strong presentation and writing skills, along with some experience with data mining and analysis. Ji Won has some experience with traditional computer vision, in the realm of handwriting recognition, but little to no experience with deep learning or neural networks. Ji Won and Kelly both have working experience as software developers in industry of 3 years.

## 1.2 Project Idea

This work aims to distinguish DeepFakes from real videos. DeepFake has been misused since it was proposed. It is difficult to distinguish them with the human eye and traditional computer vision methods since deepfakes become more powerful when deep learning develops. We will formulate the problem as a binary classification task and use deep learning methods such as convolution neural networks (CNNs) and/or recurrent neural networks (RNNs) to overcome this issue. Specifically, the input of our neural network will be a video or a frame with their corresponding labels (e.g., real or fake) as ground truth, and the output will be prediction (real or fake). Fig. 1 shows the architecture of our classification model. We will use the binary cross-entropy function (see 1) as our objective function. We hope that the neural network can implicitly capture the artifacts. For this project, we want to focus on visual deepfakes instead of voice. If we have remaining time, we would like to work on hand-crafted features such as gaze information to help the neural network explicitly capture the artifact.

$$\ell_{cls} = -\frac{1}{n}\sum_{i}^{n}[y_i log(p_i) + (1-y_i) log(1-p_i)] \tag{1}$$

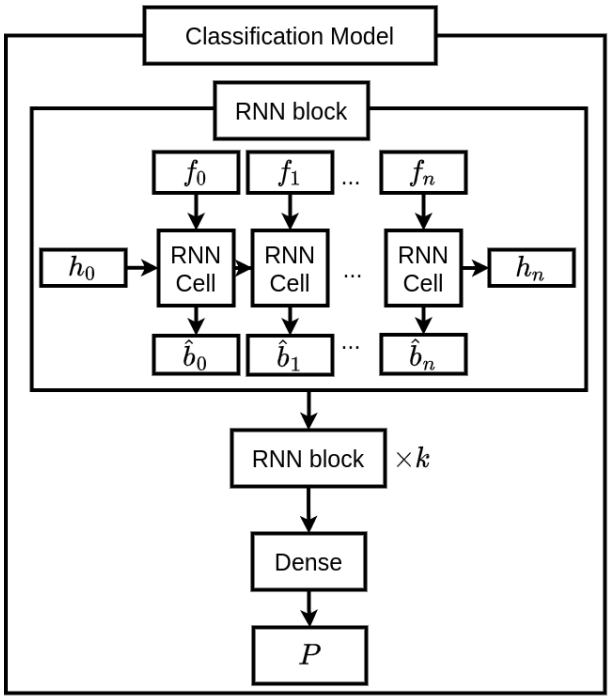


Figure 1: Our RNN deepfakes classification model

### 1.3 Data

We will be using a dataset provided by Kaggle. The full training set is 471.84 GB available in two forms: a large file containing all the samples and a set of 50 smaller files, each about 10 GB in size. The data itself is comprised of .mp4 files which have been sorted into compressed sets of 10 GB each. With each set of .mp4 files, there is a metadata.json that contains the filename, label (Real or Fake), original, and split information. There is also a set of test videos provided by Kaggle which contains 400 files (.mp4). Note that a deepfake could be either a face or voice swap (or both) in the Kaggle dataset, but we will only process the visual data.We only have limited computing resource, so we will not use the full training set. In addition, the size of our validation and test set will be 10% of our training set. A deepfake could be either a face or voice swap (or both).

Source: Kaggle

If we have enough time, we may use the FaceForensics++ dataset or Deep Fakes dataset. The FaceForensics++ dataset includes 1,000 real videos and 4,000 fake videos, and the Deepfakes dataset contains 140 wild deep fakes data.

### 1.4 Software/Hardware

For software, we plan on using Python with opencv, scikit learn, and Tensorflow 2.0. For hardware, we plan to use Google Cloud Platform (GCP) Credits to train our model.

### 1.5 Work Delegation

We will more or less split the work evenly among the three teammates, but because Cheng-You has the most previous experience on deep learning, the rest of the teammates will follow his guidance in implementing and designing the architecture. Since Ji Won and Kelly have more coding experience, they will focus on implementation. The training for batch jobs, parameter tuning, code reviews, presentation prep, and familiarization with packages will be done by all teammates.

### 1.6 Progress

We will measure the accuracy with our model on the test set. If the performance is greater than 50%, then it marks progress since it is better than random guessing. We are going to focus on how we implement the classification, and we hope to have good accuracy, but we recognize that this is a difficult task and may not receive highly accurate results.

There will be intermediary milestones:

- implement face detection via code
- binary classification of deepfake/fake videos (on a small sample)
- improve accuracy via tuning

### 1.7 Foreseen Problems

We can expect that the performance of the CNNs such as ResNet is limited since it does not fully utilize the temporal information from the video. A solution is to use RNNs such as long short-term memory (LSTM) or gated recurrent units (GRU) to capture temporal information between frames. However, the issue is that each RNN block needs to wait for hidden states, so we cannot build a huge neural network. One possible solution is to apply the self-attention mechanism in our CNNs so that it can get temporal information parallelly. We may work on this if we have extra time.

We also realize that not everyone has the same skill levels/experience with deep learning or neural networks, so there could be a huge learning curve and we may only be able to implement the code for binary classification but may not be able to improve accuracy via tuning.

### 1.8 Socio-Historical Context

Since 2017, deepfakes have been on the rise in mainstream media and have been developing rapidly in terms of technological sophistication and societal impact. With effects as large as destabilizing political impacts to as small as pranks, deepfakes have added to the degree of misinformation that is so proliferated in our society today. Source: DeepTrace

### 1.9 Benefits of this Technology

As aforementioned, deepfakes cause many issues and mistrust in society. We aim to create a technology that can start to combat this issue. While we understand that the grand scheme of this problem might be too large to tackle in the span of a final project, we hope to provide value by developing a tool that the general public can use to assess real vs fake information.

### 1.10 Potential Misuse

Studies on deepfake detection can be re-purposed to improve deepfake models, which is contrary to what we want to do. Adversaries could also find holes in deepfake detection algorithms to provide misleading results.

Additionally, we believe advertising agencies can pay celebrities less by using deepfakes; they can buy the right to use celebrity images or voices in deepfakes instead of asking celebrities to perform in person to save labor costs.

### 1.11 Extra Help

It would be great if we can access more computing resources such as GCP.