Final Project Update

By now, you've submitted your proposal for your final project and should be getting started with data collection, implementation, etc. The final project deliverable will be a website (e.g template) that will thoroughly describe the scope of your project, experiments, and results. Have your website/code repository on GT github and keep your repository public. (An alternative is to have a website via https://sites.gatech.edu/). To help aid in your final project deliverable, your final project update will be on the same website. We highly suggest you get started with a barebones website soon (like the template we provided) so you can simply worry about updating information as you make progress. Below are the subsections and requirements for the Final Project Update.

The final project update will be graded on these 5 sections:

- **Introduction/Problem Definition:** Provide a brief introduction to your project topic, and describe why it's an interesting topic to investigate. This is where you want to describe the problem itself and the motivation behind tackling it.
    - *Note: Please include 1-3 paragraphs for this section. Diagrams, visuals, and/or showing example data can be highly effective to aid in telling your story.*

  The main purpose of the project is to classify from video data whether an individual is lying and to extract key frames from the video and analyze the extracted expression to achieve algorithmic explainability. The lie detector AI aims to work as a highly accurate lie detector, which has significant application in a judicial or crime investigation scenario. The lie detector AI has the potential to determine innocent or guilty cases without bias or discrimination. However, such use cases require a concrete base of evidence, hence, we plan to extract key expressions by combining expression analysis to provide logical reasoning based on the model's prediction. Ultimately, we are utilizing computer vision techniques to pre-process video data, extract visual features, and pass them into vision transformer-based video classifiers to classify facial expressions.

  Detecting when a person is lying through a rigorously-tested AI model can play a significant role in judicial and legal settings, where an individual's demeanor and facial expressions can indicate whether or not they are telling the truth. From an objective standpoint, the AI is highly unlikely to discriminate or bias, but rather classify whether or not the defendant (or accuser) is lying when under question. Regarding a criminal case, it is immediately apparent the advantages of a lie detector in determining the innocence or guiltiness of the accused in a court of law. Theoretically, the lie detector AI can play a considerable role in any political,

economic, or medical institution, including holding a presidential candidate accountable to the information they may claim in a debate or the credibility of any doctors when under question of their practice.

- **Related Works:** Describe related works in your problem space (research papers, libraries/tools, etc) for existing solutions for this problem or adjacent areas. Make sure to cite papers you reference!
    - *Note: Related work should have between 2 to 6 sentences for each work citing. Please cite works following IEEE guidelines. Organize related work into different subsections based on similarity of approaches.*

        - ViViT:

        Uses transformer architecture to achieve video classification by passing in temporal frame patches. Specifically, the model uses three variants of attention to factorize the spatial and temporal dimensions of the input. It enables the model to regularize effectively even if the training data is not large enough.

        Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., & Schmid, C. (2021). Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6836-6846).

        - Deep Lie: Detect Lies with Facial Expression:

        This Stanford paper uses a computer vision-based approach to detecting lies in video streams, improving upon previous work on the same subject from 2017. Compared to previous work, the model proposed in this paper trains a separate facial expression recognizer to capture key frames with pronounced expressions.

        Feng, K. (2021). DeepLie: Detect Lies with Facial Expression (Computer Vision). CS230: Deep Learning, Spring 2021, Stanford University, CA. https://cs230.stanford.edu/projects_spring_2021/reports/0.pdf

        - Lie Detector AI: Detecting Lies Through Fear:

        This paper uses OpenFace software in conjunction with a random forest classifier to classify videos into truthful or deceitful. Following a CNN with a random forest classifier improves the accuracy from 52% with only CNN to 86%.

Kafadar, V. A., Ullmann, L., Haudenschild, J., Kafadar, V. A., & Tschakert, H. (2022). Lie Detector AI: Detecting Lies Through Fear. https://doi.org/10.13140/RG.2.2.18265.19040

- Facial expression recognition using CV:

The study focuses on the classification of emotion based on human facial expressions. Methods included face detection, PCA, smoothing (pre-processing), Optical flow (feature extraction), and Gabor filters. Includes various datasets for facial expression recognition.

Canedo, D., & Neves, A. J. R. (2019). Facial Expression Recognition Using Computer Vision: A Systematic Review. Applied Sciences 2019, Vol. 9, Page 4678, 9(21), 4678. https://doi.org/10.3390/APP9214678

- Face-Focused Cross-Stream Network for Deception Detection in Videos:

This study implemented both face expression and body gestures for deception detection. A face detection algorithm detected the roi for the face and the rest of the frame was used for motion detection and tracking. A correlation learning was performed for joint deep feature learning from face expression and motion. They also added adversarial learning and meta learning to ameliorate for lack of training data.

Ding, M., Ding, M., Ding, M., Zhao, A., Zhao, A., Lu, Z., Lu, Z., Xiang, T., Xiang, T., Wen, J.-R., Wen, J.-R., Wen, J.-R., & Wen, J.-R. (2019). Face-Focused Cross-Stream Network for Deception Detection in Videos. Computer Vision and Pattern Recognition. https://doi.org/10.1109/cvpr.2019.00799

- Unmasking the Devil in the Details:What Works for Deep Facial Action Coding?:

This study explored detection of facial action unit occurrence and estimation of facial action unit intensity using deep learning. Effects of different components and parameters such as normalization, model architecture, training set size, optimizer and learning rate on model performance were investigated.

Niinuma, K., Jeni, L. A., Onal Ertugrul, I., & Cohn, J. F. (n.d.). Unmasking the Devil in the Details: What Works for Deep Facial Action Coding?

https://bmvc2019.org/wp-content/uploads/papers/0403-paper.pdf

- **Methods/Approach:** Indicate algorithms, methodologies, or approaches you used to craft your solution. What was the reasoning or intuition for trying each methodology/algorithm. What does the overall pipeline look like and the details behind each component? Make sure to establish any terminology or notation you will continue to use in this section.

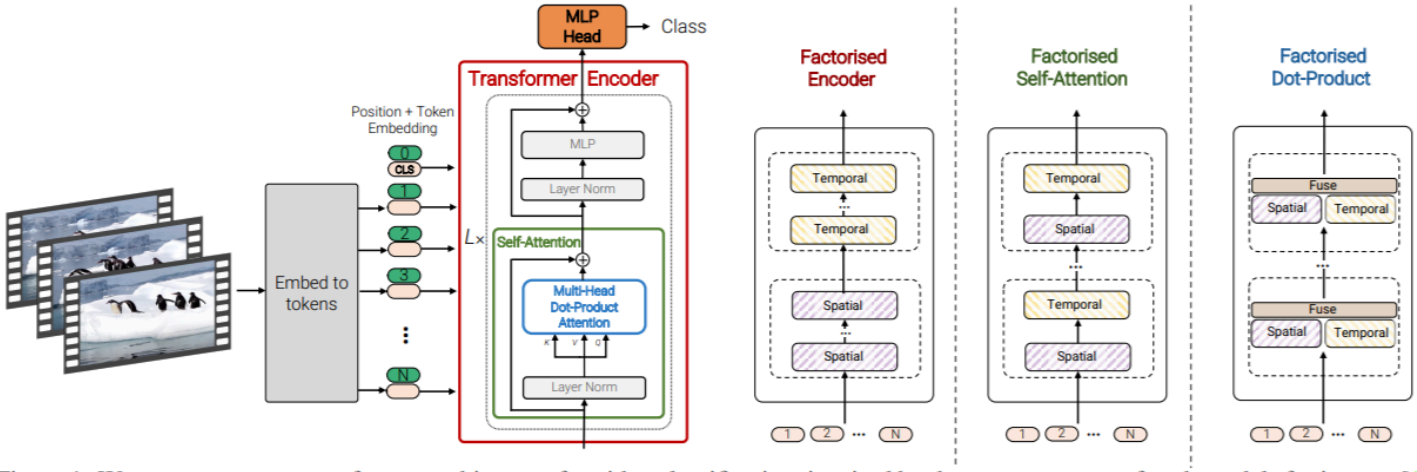

Figure 1: We propose a pure-transformer architecture for video classification, inspired by the recent success of such models for images [15]. To effectively process a large number of spatio-temporal tokens, we develop several model variants which factorise different components of the transformer encoder over the spatial- and temporal-dimensions. As shown on the right, these factorisations correspond to different attention patterns over space and time.

Methodology:

ViViT Architecture:

The ViViT model first extracts spatial-temporal tokens from the input video by embedding, which are then encoded by a series of transformer layers. The model then uses several variations of attention to factorize the large spatial and temporal dimensions of the video. Specifically, it uses a factorized encoder layer, a factorized self-attention layer, or a factorized dot-product layer. In our case, we primarily use a factorized self-attention layer. This allows us to effectively train and regularize the model when the datasets are not large enough.

Problem with sampling key frames in ViViT:

While ViViT achieved good results on video classification, however, the two methods used to sample frames --1) uniform sampling: Uniformly sampling picture frames from the video with fixed time span and dividing into image patches; 2) Tubelet embedding: selecting fixed time span of patches in frames into embeddings, are both arguably flawed for capturing transient frames such as micro expressions. Uniform sampling has no guarantee of capturing the exact

high-relevance frames, and Tubelet embedding can result in low attention weights of tubes containing high-relevance frames because of additional low-relevance expressions. As a result, we would like to improve the sampling process to obtain more relevant key frames.

Improving key frame sampling through facial action unit tracking:

Instead of sampling key frames by uniform sampling and Tubelet embedding, we use a technique called Facial Action Unit (FAU) tracking. A FAU is a term used in facial expression analysis and the study of human facial movements. It was introduced by psychologists Paul Ekman and Wallace V. Friesen as part of their Facial Action Coding System (FACS). We can calculate FAU intensity using facial landmarks. Higher FAU intensity means higher likelihood of micro expressions. In the context of lie detection, micro expressions can be a good indicator of whether the person is lying. Thus, by sampling key frames with high intensity, we can capture frames more relevant to the prediction.

- **Experiments / Results:** Describe what you tried and what datasets were used. We aren't expecting you to beat state of the art, but we are interested in you describing what worked or didn't work and to give reasoning as to why you believe so. Compare your approach against baselines (either previously established or you established) in this section. Provide at least one qualitative result (i.e. a visual output of your system on an example image).
    - *Note: For the project update, feel free to discuss what worked and didn't work. Why do you think an approach was (un)successful? We expect you to have dealt with dataset setup and completed <u>at least 1</u> experimental result by the project update.*

Datasets: We made use of the 'Miami University Deception Detection Database' that contains 300+ videos of interviewees speaking of truthful and lying statements that come with demographic information, sentiment analysis and transcripts. We selected it because it has good size and was used in the original research team's study which returned good results. The video frame is also clean, which has the person's face occupying the center of the frame.
Framework: The video is processed through FAU, then the high intensity frames are extracted for the ViViT model, each video sample is represented as a 4-dimensional array [frame_num, height, width,channel], where frame numbers are padded to the maximum length. By using the vision transformer architecture, we are allowing inputs of different temporal lengths, which makes the framework flexible. Lastly, the ViViT outputs a percentage prediction that predicts the veracity label of a given video.
Experiments & Results: At the moment, we are testing the framework on 61 video samples for the purpose of speeding up and saving computational resources. The FAU extracted a maximum sequence length of 105 frames per video.

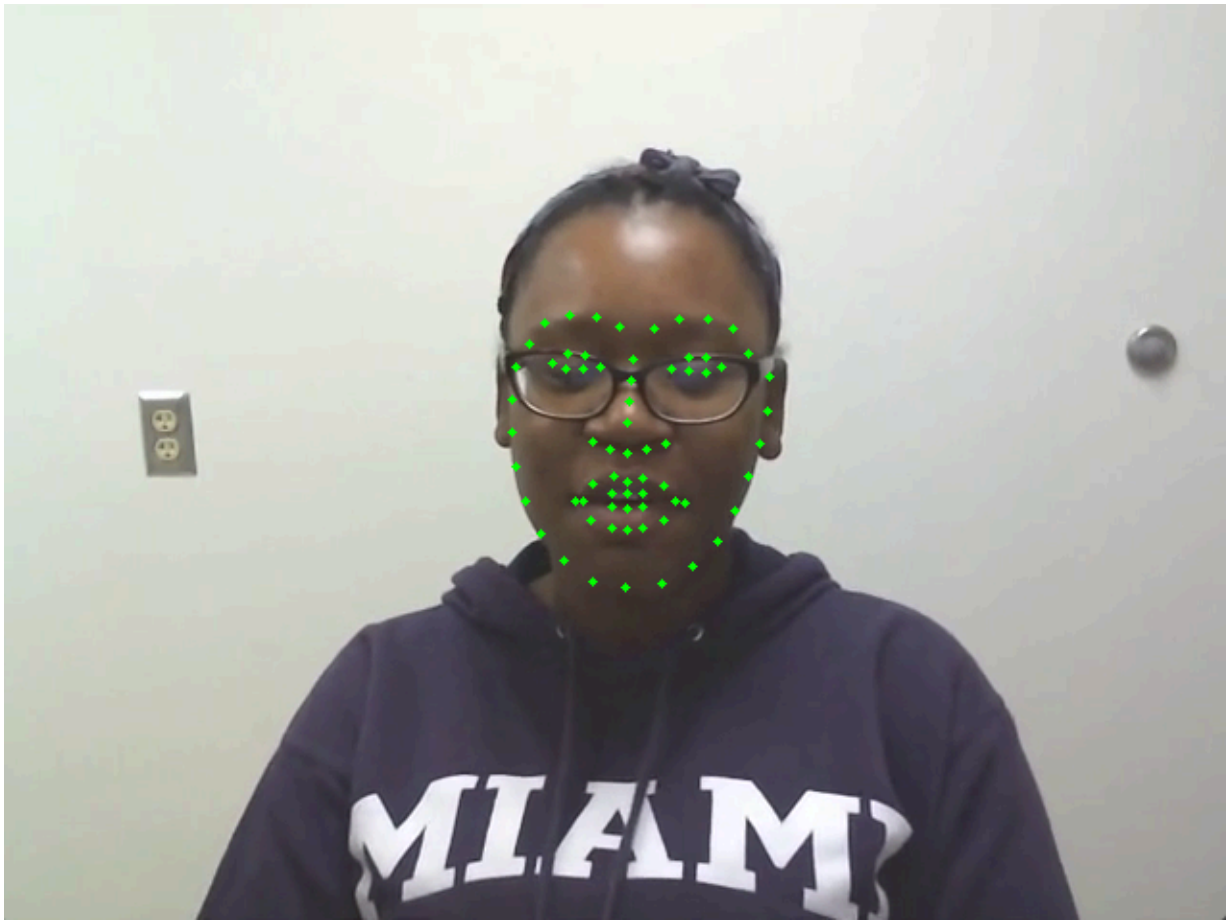

Figure 2: 68-point facial landmarks for facial action unit (FAU) calculation

The model uses Adam optimizer and binary cross entropy loss function with 0.001 learning rate, running 20 epochs, achieving steadily plateaued training loss and reaching a low 57.14 % accuracy. However, by tweaking the training sample size, we have discovered that increasing training size increases the model's performance steadily as shown in the data below (tested on the same 10% test data). Combining the fact that transformers are more data hungry, we believe the model shall have significant performance boost given the full dataset.

Training Data Size	70%	80%	90%
Test Accuracy	49%	51%	57%

- **What's next:** What is your plan until the final project due date? What methods and experiments do you plan on running?
    - *Note: Include a task list (can use a table) indicating each step you are planning and anticipated completion date.*

Baseline Comparison: Our work focuses on having a case-specific use of the vision transformer technology. Therefore, we will be setting the ViViT using the original even temporal sampling method trained on the same dataset as our baseline model.
Model Optimization: The model is functional and yields moderate results (57% accuracy), however, the model theoretically should yield significant outcomes for higher accuracy and lower cross-entropy loss. Hence, the work to be done mostly consists of enhancing model performance along with baseline comparison.
To enhance its performance, we plan to tweak the model's hyperparameters to optimize prediction and yield better results. Other work includes training on the full dataset, early model stoppage, saving/loading optimal model runs and inference.
The baseline will be trained on evenly sampled temporal frames and tested in comparison with the proposed model.
Explainable AI: The explainable AI will be achieved by extracting attention weights from corresponding frames and adding sentiment analysis of the extracted frames to provide interpretability.

Task	Methods	Anticipated completion	Member
Process videos (Calculate FAU)	FAU calculations from facial landmarks	04/03/24	Farhan
Extract key frames (from FAU intensity peaks)	Signal processing and using derivative filters	04/10/24	Farhan
Train on all samples (Proposed & Baseline)	DataLoader	04/12/24	Vikram
Early model stoppage (setting training callbacks)	PyTorch - EarlyStopping (score_function)	04/14/24	Vikram
Tuning hyperparameters	Adjust hidden layers, neurons per layer, learning rate, batch size	04/14/24	Cheting
Save/load models	PyTorch: Save model if accuracy is higher than best, loss lower than best	04/14/24	Vikram
Extract attention weights	PyTorch transformer	04/03/24	Cheting
Sentiment Analysis	Deep learning approach	04/20/24	Yiheng

Inference function (Simple UI)	Model output, labels and the extracted frames	04/20/24	Yiheng

- **Team member contributions**: Indicate what you anticipate each team member will contribute by the final project submission.
    - *Note: List every member name and their corresponding tasks in bullet points – or you may simply assign team member names to the task list you created above.*

        - Cheting Meng: Build dataloader, Model architecture, Model training, Experiments

        - Vikram Sagar: Introduction/Problem definition, train/test data split and model training

        - Yiheng Mao: Related work, Methodology documentation

        - Farhan Khan: FAU calculation, video processing, peaks detection from FAU intensity timeline, key frame extraction