Chuck de Sully

Final Project – Blinking Morse Code

Project Description:

This project has a goal to develop a solution aimed at detecting human blinks and translating these gestures into Morse code, subsequently decoded into letters and words. This challenge addresses the need for alternative modes of communication, particularly in scenarios of severe physical limitations. For instance, individuals with conditions such as quadriplegia, if unable to speak, may still possess the ability to communicate via eye movements, thereby potentially establishing vital connections with the external world. Inspiration stems from a desire to empower individuals facing extreme handicaps, elevating their capacity to engage meaningfully with others.

Although simplistic in nature, this project exemplifies the potential implications extending far beyond blink interpretation alone. By establishing a robust framework capable of deciphering various bodily neural responses into actionable communication, we pave the way for a broader understanding of non-verbal language cues. This not only touches on linguistic interpretation but also opens avenues for the execution of physical actions, thereby enhancing the autonomy and agency of individuals facing debilitating challenges.

Objective:

The primary objective is the development of a responsive model that operates in real-time, devoid of excessive resource demands. It is important that the solution provides instantaneous feedback, enabling live interaction within a dynamic environment. The desired outcome entails the immediate interpretation of each Morse-encoded letter, bridging the gap between gesture and language in real-time scenarios. These objectives are intricately aligned with a goal of facilitating real-time communication for individuals grappling with physical limitations.

Datasets:

For training, a small custom dataset was self-curated comprising of stock images of faces exhibiting both open and closed eyes. To ensure diversity, the dataset included individuals of varying demographics, encompassing all genders, different age groups, and a range of ethnicities. Additionally, we captured images of our own face to enrich the dataset.

Preprocessing was a crucial step in preparing the dataset for training. All images were converted to grayscale and uniformly cropped to a standardized 120x30 pixel frame, capturing both eyes along with a slight buffer space around them. This consistent framing ensured uniform input during evaluation inference, enhancing model robustness.

To augment the dataset and introduce variability, we applied further transformations during training. These included random horizontal and vertical flips, adjustments to brightness and contrast, and random rotations of up to 10 degrees. These transformations enabled the model to learn from a broader range of patterns, despite the small size of the dataset.

Modeling Approach:

The initial step in the modeling approach involved simplifying the input frames by converting them to a single channel of grayscale. This decision was based on the premise that color information is unnecessary for discerning whether the eyes are open or closed.

Subsequently, there was focus on isolating the eye regions within the images. Leveraging the dlib package, eye landmarks were located. This information enabled the process to rotate and crop the images around the eyes, ensuring that regardless of facial orientation, the eye inputs remained horizontally aligned and free from extraneous facial features.

During research, approaches to utilize dlib for blink detection based on landmark distance ratios came up. The goal with this project is to utilize PyTorch for more robust detection as there is significant variability in landmark positions and ratio thresholds across individuals.

The modeling architecture is primarily comprised of convolutional layers, with just two layers being adequate. The first convolutional layer employed a kernel size of 5, followed by a second layer with a kernel size of 3. Each convolutional layer was succeeded by a MaxPool2d layer with a kernel size of 2.

The outputs from the convolutional layers were then used as input into two linear layers. The first layer, referred to as fc1, had an output size of 128, while the final layer, fc2, had an output size of 2. Our objective was to train fc2 using CrossEntropyLoss to accurately classify the two categories of "open" and "closed" eyes. The goal was to freeze the weights of the model and utilize fc1 as an embedding for a time series model, as inspired from a previous class assignment.

The original idea was to explore and to produce a long-term recurrent convolutional network (LRCN), with a time series element of either a Long Short-Term Memory (LSTM) network or a transformer encoder. However, these attempts encountered challenges, with the LSTM-based approach failing to achieve real-time inference, and the transformer model struggling to surpass an accuracy of 30%. It was decided to adopt a simpler approach, leveraging the standard timings of Morse code to interpret blinks and map them to an indexed alphabet.

A key aspect of our simple convolutional network architecture lies in the joint training of both fc1 and fc2. Initially, training focused on fc2 to accurately classify the two eye states. Next, a k-means clustering algorithm was applied to identify two embedding centers based on all observed pattern outputs of fc1. Further training was then conducted on fc1 to minimize the distance to these centers, facilitating enhanced classification performance.

Experimental Results:

Experimentation revealed a notable pattern that underscored the rationale behind adopting a sequential training methodology. Initially, while training fc2 to classify eye states, a correlation matrix was computed from the outputs of fc1. As anticipated, most patterns exhibited strong correlation within their respective classes. However, certain training patterns displayed suboptimal correlation scores with their class counterparts, and continued training on fc2 appeared to exacerbate this discrepancy.

Upon the generation and training of embedding centers, these problematic patterns exhibited convergence towards similar representations. Consequently, the initial training of fc2 established distinct differences between classes, while subsequent training of fc1 solidified the positioning of embeddings. This strategic approach set the stage for new patterns encountered during inference to have a better likelihood to align with specific embedding centers, enhancing classification accuracy.

Interestingly, simultaneous training of fc1 and fc2 did not yield superior results compared to sequential training. This finding suggests that the sequential approach offers a more effective strategy for refining the model's representation and classification capabilities.

In the exploration of alternative approaches, empirical tests were conducted using dlib and facial landmarks for blink recognition. However, findings indicated that dlib was not as reliable as the convolutional approach, further emphasizing the efficacy of this project’s chosen modeling methodology.

Following are some stats regarding the sequential training which include the very tiny testing data and then the complete data for comparison.

Upon 500 epochs of classification training:

fc2 training data best accuracy : 95.7%

|  |  |  |
| --- | --- | --- |
| Test Data | Fc2 classification | Fc1 clustering |
| Accuracy | 80% | 60% |
| F1 Score | 0.762 | 0.375 |
| Precision | 0.875 | 0.3 |
| Recall | 0.75 | 0.5 |
|  |  |  |
| Training+Test Data | Fc2 classification | Fc1 clustering |
| Accuracy | 96.43% | 71.43% |
| F1 Score | 0.964 | 0.689 |
| Precision | 0.967 | 0.818 |
| Recall | 0.964 | 0.714 |

Upon 500 epochs of cluster categorization training:

Fc2 training data best accuracy: 100%

|  |  |  |
| --- | --- | --- |
| Test Data | Fc2 classification | Fc1 clustering |
| Accuracy | 80% | 100% |
| F1 Score | 0.8 | 1.0 |
| Precision | 0.833 | 1.0 |
| Recall | 0.833 | 1.0 |
|  |  |  |
| Training+Test Data | Fc2 classification | Fc1 clustering |
| Accuracy | 92.857% | 100% |
| F1 Score | 0.928 | 1.0 |
| Precision | 0.938 | 1.0 |
| Recall | 0.929 | 1.0 |

As can be seen, in the second phase of training by cluster center minimization, there was a slight negative impact on the fc2 classification scores, but it is overshadowed by the fact that determining categories by which cluster center the fc1 embedding data is closest to received 100% accuracy in both training and testing data. Albeit exciting, this dataset still suffers from lack of numbers and with further study will need to increase the sample size to have critical success in real life scenarios.

Conclusion:

This project encountered several challenges that warrant consideration. Firstly, while dlib proved effective in landmarking eyes within a full-face context, its limitation lies in the requirement for the entire face to be visible for accurate landmarking. Consequently, this project is constrained to scenarios where the camera can capture the full face, limiting its applicability in certain contexts. Additionally, the project's focus on a single face poses challenges in scenarios with multiple faces present, as the current implementation does not account for multiple eye landmarks simultaneously.

Secondly, an attempt to implement a Long Short-Term Memory (LSTM) Recurrent Convolutional Network (LRCN) variant encountered hurdles due to the nature of Morse code sequences. The inherent variability in sequence lengths posed difficulties in producing uniformly sampled datasets, essential for batching using the PyTorch DataLoader. While one solution involved padding shorter Morse code videos to match longer ones, this approach proved counterproductive, making training exceedingly challenging. A batch size of one was adopted and the training loop restructured to optimize loss accumulation and backward propagation, simulating a batch of the full data set, achieving better training stability.

Lastly, a notable disparity arose between training sequences of patterns and conducting live inference. The inherent ambiguity in live inference, particularly regarding the position of frames within a Morse encoding sequence and not knowing if a frame is at the end of a short sequence or in the middle of a long sequence, led to unpredictable predictions. This challenge underscores the need for further exploration into real-time inference strategies to address the dynamic nature of Morse code interpretation in a live setting.

The end result did have a very lightweight neural network footprint and could show promise on portable devices.