Hand Gesture Recognition for Video Conferencing

CALUM SIEPPERT (30093813), University of Calgary, Canada

This paper describes a project aimed at developing an image classifier that can identify hand gestures commonly used in video calls: thumbs-up, thumbs-down, okay, and no gesture. The project utilizes the HaGRID dataset and two neural network architectures: MobileNetV3-Large and AttentionHGR. The former is a well-established, lightweight architecture designed for efficient image classification tasks, while the latter is a more complex architecture that utilizes residual attention modules for better feature extraction. The project aims to provide a real-time, on-device solution for identifying hand gestures during video calls without requiring participants to share their camera feeds. The results show that both models achieve high accuracy rates, with AttentionHGR outperforming MobileNetV3-Large. The conclusion is that both architectures can be used for hand gesture recognition, while AttentionHGR's residual attention modules may provide better classification performance, and MobileNetV3's efficient architecture may provide better performance for this particular application.

Additional Key Words and Phrases: neural networks, hand gesture recognition

ACM Reference Format:
Calum Sieppert (30093813). 2023. Hand Gesture Recognition for Video Conferencing. 1, 1 (April 2023), 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

INTRODUCTION

This project aims to develop an image classifier which identifies the following hand gestures in a video call participant's video feed: "thumbs-up", "thumbs-down", and "okay". The project also aims to classify the case of none of the aforementioned gestures being made.

Such a classifier can be used during online video calls to increase interaction and engagement without requiring the participants to share the feed from their cameras. For example, at the University of Calgary, cameras are rarely "on" apart from the professor's, which can increase the difficulty of teaching as the professor receives no feedback from the students. As another example, individuals who work remotely do not always share their camera during meetings, which can again make communication more difficult. It is also not always feasible for large meetings to have everyone's camera on due to bandwidth issues. Furthermore, in large meetings it isn't feasible to gauge the reactions of everyone by looking at their individual video feeds.

With a classifier for common "reaction" gestures, video call participants could easily and naturally provide feedback during a video call. There are often emoji "reactions" built into video call software, however they are slow and unnatural to use, which does not allow for the type of natural real-time feedback that people are used to in their interactions. This project's classifier can provide a compromise between the clear desire to not always share camera feed, and the need for real-time natural feedback in video calls.

The project will utilize the HaGRID dataset [2], a large hand gesture recognition dataset. Kapitanov et al. also provides various benchmark models for the hand gesture classification task. Notably, MobileNetV3-Large [1] achieves very good performance in both classification scores and speed, therefore it is well suited for this project.

Miah et al. is a more recent paper presenting a novel architecture for hand gesture recognition inspired by the FusAtNet [4] architecture. The architecture utilizes residual attention modules to build spatial features and highlight key areas of the image, which is well suited for hand gesture recognition as the hand gesture in a given image may only be one small part of the image. This architecture will be referred to as AttentionHGR.

MATERIALS AND MODEL ARCHITECTURE

MobileNetV3-Large

MobileNetV3-Large is a convolutional neural network architecture designed for efficient image classification tasks on mobile and embedded devices. It was introduced by Google AI researchers in 2019 as an improvement over the previous versions of MobileNet. The architecture is shown in Figure 1.

The MobileNetV3-Large architecture uses several techniques to reduce the computational cost and improve the accuracy of the network. It utilizes a combination of depthwise separable convolutions and linear bottlenecks, which reduces the number of parameters and computation required while maintaining the expressiveness of the model.

It also uses a squeeze-and-excitation module, which selectively amplifies important features in the input. Additionally, the network has a hard-swish activation function, which is a faster and more accurate alternative to the traditional ReLU activation function.

MobileNet was chosen due to it's good performance in the benchmarks for the Hagrid dataset, it's ability to be trained on my laptop's CPU, and it's speed and size which are well suited for real-time on-device hand gesture recognition. No modifications were made to the architecture, except to replace the head with a single fully connected layer with 3 outputs.

Fig. 1. MobileNetV3-Large architecture diagram (Pytorch's implementation visualized by tensorboard)

Author's address: Calum Sieppert (30093813), calum.sieppert@ucalgary.ca, University of Calgary, Calgary, Alberta, Canada.

XXXX-XXXX/2023/4-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

AttentionHGR

The model expects 160x160 RGB images as input. The model consists of two "attention" stages and a classification stage. In the attention stages, feature extraction and attention modules are combined. The original image is concatenated with the output of the first stage, so the second stage has the output of the previous layer and the image data as input, with the idea being that the first stage extract spatial features from the image to aid the second stage in highlighting key areas of the image.

The feature extraction modules are basic six-layer 2D convolutional networks with each layer followed by ReLU activation and batch normalization. The attention modules are six-layer convolutional networks with residuals after the second and fourth layers, and again each convolutional layer is followed by ReLU activation and batch normalization. Element-wise multiplication is used to combine the concurrent attention and feature extraction modules. Finally, the output of the second stage is fed to a fairly standard convolutional classifier module utilizing ReLU activation, batch normalization, max pooling, and softmax for the final output. The kernel size of all convolutional layers is 3x3.

This architecture was chosen due to its good performance on similar datasets, and for educational purposes as I wanted to practice implementing and training a complex architecture. The attention modules in theory are well suited for the task of hand gesture recognition as the gesture will likely make up only a small part of the image. Figure 2 displays the original architecture for AttentionHGR.

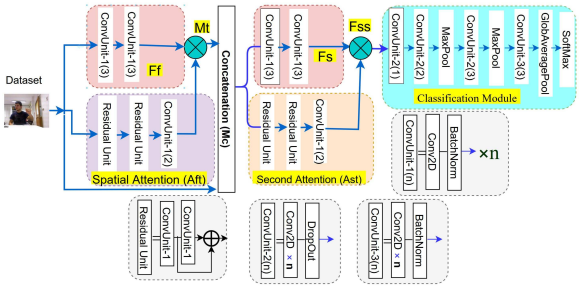

Fig. 2. AttentionHGR flow diagram

The full architecture as described in the original paper was modified for this project's task. In particular its size was significantly reduced to align better with the simplified task of only classifying three gestures, and to reduce training time. Focusing on the most successful iterations:

- "arch-shrink-3": classification module reduced to only have two convolution layers with a final linear activation layer. Feature extractors reduced to have only 2 convolutional layers with fewer filters. Attention modules reduced to have only one residual unit followed by two convolutional layers, again with fewer filters.
- "arch-shrink-4": same as arch-shrink-3 but with even fewer filters all around, only one convolutional layer at the end of the attention modules, and only one convolutional layer in the classification module. Figure 3 displays this architecture.

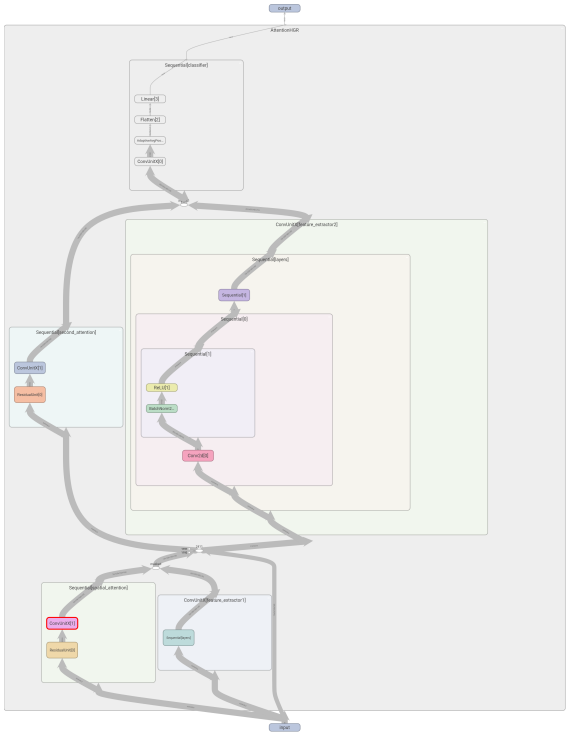

Fig. 3. arch-shrink-4 architecture diagram (Visualized by tensorboard)

DATASET

The Hagrid dataset [2] was designed for hand gesture recognition systems. The dataset is the result of crowd-sourcing in Russia, and it is approximately 716 GB including more than 550 thousand high-resolution images divided into 18 classes of gestures. For this project only a subset of the dataset is used, including only the gesture classes "like", "dislike", and "ok". The classes are fairly evenly distributed, with each class containing between 30-32 thousand examples.

The dataset contains many unique faces and scenes. The subjects are people from 18 to 65 years old. The ratio of women to men is about 27 to 20. The dataset was collected mainly indoors with considerable variation in lighting, including artificial and natural light. The dataset includes images taken in extreme conditions such as facing and backing to a window. The subjects also had to demonstrate gestures at different distances from the camera.

Data on the distribution of skin-tones of the subjects is not available, but given the singular location of the crowd-sourcing and some scrolling through the images, the dataset does not appear to contain much diversity of skin-tone. Another possible bias is towards particular users (subjects of the images): if the dataset is not stratified by user, the model may become biased towards certain users instead of generalizing. Finally, the dataset only contains single subject images (one person per image), however this is suitable for the task as most video conferencing is done with one person per camera.

Apart from stratification by user, to deal with storage space limitations the images are resized on the file system. This reduces the

storage requirement from >90GB to <1GB, and allows the data to be uploaded to Google Colab. For pretrained MobileNet, images are resized, center-cropped, and normalized to match the pretraining dataset. For AttentionHGR, so far only image resizing is used, as the original paper does not appear to have utilized image augmentation, and the performance achieved thus far has not reached that of the original paper. Image augmentation could be experimented with once the performance is closer to the original paper.

Figure 4 shows the types of images in the dataset, and figure 5 shows the different hand gestures.

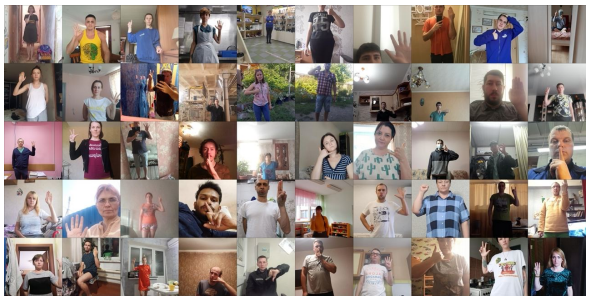

Fig. 4. A sample of images from Hagrid

Fig. 5. Classes of Hagrid

TRAINING PROCEDURE

All training is performed using Pytorch. MobileNet was trained on CPU, AttentionHGR was trained on unpaid Google Colab GPU.

For MobileNet, stochastic gradient descent is used with learning rate 0.001 and momentum 0.9. I utilized the pretrained weights provided with Pytorch, froze them initially to avoid the random gradients from the classification head clashing with the pretrained weights, then unfroze the whole model for the rest of the training as the dataset was very large. The model was trained on the entire dataset with batch size 512, and 90% of the data used for training. The classification head was kept simple to help avoid overfitting. After initially freezing, the entire model was unfrozen and using the Adam optimizer with learning rate 0.0001 and weight decay 0.0005 it was trained for five epochs.

For AttentionHGR, due to the complexity and size of the model, an iterative approach is taken to training. For a given variation of the architecture, the architecture is first validated by training on a single batch (because the model is implemented from scratch in Pytorch, this is a good way to check for mistakes). Then the model is trained on a subset of the dataset (usually 1500 to 4500 images), with 10-30% validation set sizes. The best performing model for the subset would ideally be trained on the entire dataset, but it is unlikely that this can be done with unpaid Google Colab. The architecture variants are trained using Adam optimization with Nesterov momentum. Variants with a final LogSoftmax layer are trained with negative log likelihood loss, otherwise they are trained with cross entropy loss. Variants are trained with learning rates varying from 0.000005 to 0.0001, with some incorporating weight decays of 0.005 to 0.0005 to help with overfitting. Batch size ranges from 4 to 16 depending on model size (which determines remaining free GPU RAM). Tensorboard is used to monitor training progress, compare architectures, and watch for plateaus in training and validation loss, displayed in Figure 6. Dropout is used in some variants in an attempt to prevent over-fitting, however this has not seen much success.

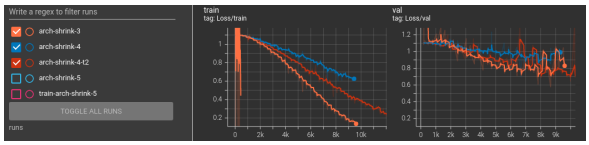

Fig. 6. AttentionHGR training (left) and validation loss (right) curves

EVALUATION PROCEDURE

Standard classification metrics are used to assess model performance: precision, recall, and F1-score. These metrics provide a good all-around evaluation of the model's performance for classification as they account for not only the overall accuracy but possible biases of the model.

For AttentionHGR, I have not yet been able to train the model to have performance comparable to MobileNet or the original paper, so the evaluation thus far has been based on validation loss and F1-score following each epoch in training. The original paper reported results of 99.75% accuracy for a similar dataset classifying 10 different hand gestures. The best validation F1-score so far has been ~73 achieved by arch-shrink-4. The model is prone to overfitting, as demonstrated by the training graphs in figure 6 showing the discrepancy between training and validation loss. Future work for this model would focus on better adapting the architecture to the task, likely by further shrinking the architecture.

For MobileNet, the model was selected based on the F1-score evaluated on 10% of the training set after each epoch of training. The model was evaluated on a separate testing set, and achieved an average of 97 precision, recall, and F1-score, which is comparable to the MobileNet benchmark provided with the Hagrid dataset.

	precision	recall	f1-score	support
like	0.94	0.99	0.96	2415
dislike	0.99	0.97	0.98	2551
ok	0.99	0.96	0.98	2391
accuracy			0.97	7357
macro avg	0.97	0.97	0.97	7357
weighted avg	0.97	0.97	0.97	7357

Figure 7 shows the confusion of the model. We can see that it struggles slightly distinguishing "like" gestures with a bias towards

the class. A few examples of errors by the model in Figure 8 demonstrate that the model is struggling on images that even humans would struggle with.

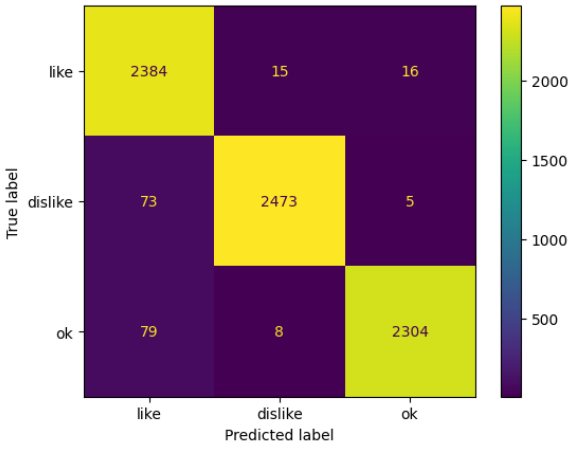

Fig. 7. MobileNetV3 confusion matrix

Fig. 8. MobileNetV3 errors

Further, the model provides (seemingly) instant predictions at 20+ frames per second on my laptop's CPU, utilizing about 250% CPU (two and a half cores). The model's predictions are generally fairly accurate (tested on various family members in different locations of my house). It struggles somewhat with certain variations, namely when multiple people are in frame, certain lighting conditions, uncommon distances of the gesture from the camera, and lack of contrast between the gesture hand and its background. The model also struggles in confidence in that it frequently switches between predictions. With application level modifications such as averaging of predictions and reduced frequency of predictions, the model would be well suited for local hand gesture classification during video calls.

Further work for the MobileNetV3 model would likely focus on data preparation, for example experimenting the data augmentation, and adapting the preprocessing to avoid images like the ones shown in figure 8.

DISCUSSION

The MobileNetV3-Large and AttentionHGR models achieved good performance on the Hagrid dataset, with MobileNetV3-Large performing better. Pytorch's MobileNet implementation achieved performance comparable to the benchmark provided for the Hagrid dataset in hand gesture classification. The project suggests that the MobileNetV3-Large architecture is well suited for real-time hand gesture recognition on personal devices due to its speed, size, and ability to be trained on a laptop's CPU. The project and related work also suggest that the AttentionHGR model's attention modules may be well suited for recognizing hand gestures in images where the gesture makes up only a small part of the image.

Further research could explore different architectures or modifications to existing architectures, such as adding attention modules to MobileNetV3-Large or exploring different attention module architectures for AttentionHGR. In particular, further modifications to the original AttentionHGR architecture may make it less prone to overfitting. Additionally, further research could investigate bias in hand gesture recognition datasets and develop methods to reduce or eliminate it, in particular focusing on providing a diversity of skin tones among image subjects. The findings could have applications in improving communication over video calls where camera sharing is either not possible or undesirable.

CONCLUSION

In conclusion, this project aimed to develop an image classifier that can identify common hand gestures, such as "thumbs-up," "thumbs-down," and "okay," in a video call participant's feed. The project utilized two architectures: MobileNetV3-Large and AttentionHGR, and evaluated their performance on the HaGRID dataset. The results showed that both models achieved good performance, MobileNetV3-Large outperforming AttentionHGR.

The impact of this project is significant, as it offers a solution to the challenge of engaging participants during video calls without requiring them to share their camera feed. By enabling real-time feedback through natural hand gestures, video calls can be more interactive, engaging, and effective. Furthermore, this project demonstrates the potential of machine learning for solving real-world challenges and improving communication and collaboration.

Future directions for this project include exploring the use of transfer learning to improve the performance of the models on new datasets and expanding the range of hand gestures that can be recognized. Additionally, the application of this technology can be extended beyond video calls, such as in the development of gesture-controlled interfaces for various devices and applications. Overall, this project highlights the potential of machine learning to enhance human interaction and communication.

REFERENCES

[1] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. 2019. Searching for MobileNetV3. *CoRR* abs/1905.02244 (2019). arXiv:1905.02244 http://arxiv.org/abs/1905.02244

[2] Alexander Kapitanov, Andrew Makhlyarchuk, and Karina Kvanchiani. 2022. HaGRID - HAnd Gesture Recognition Image Dataset. https://doi.org/10.48550/ARXIV.2206.08219

[3] Abu Saleh Musa Miah, Md. Al Mehedi Hasan, Jungpil Shin, Yuichi Okuyama, and Yoichi Tomioka. 2023. Multistage Spatial Attention-Based Neural Network for Hand Gesture Recognition. *Computers* 12, 1 (2023). https://doi.org/10.3390/computers12010013

[4] Satyam Mohla, Shivam Pande, Biplab Banerjee, and Subhasis Chaudhuri. 2020. FusAtNet: Dual Attention Based SpectroSpatial Multimodal Fusion Network for Hyperspectral and LiDAR Classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.