Real-time object detection of the ASL alphabet:

Eric Feng, Rain Juhl

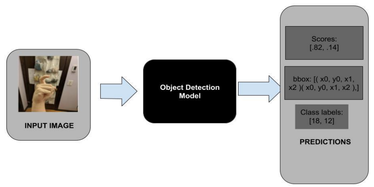
Department of Computer Science

Introduction and Summary of Works

In this paper we will experimenting with real time American Sign Language (ASL) Object Detection and classification. Nearly 48,000,000 persons are hard of hearing, with nearly 5,000,000 of those people functionally deaf. Only 2,000,000 or so americans are able to understand ASL.

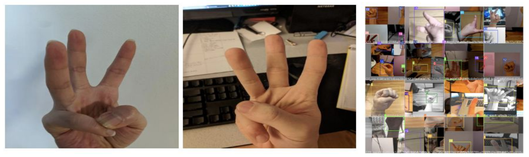
Companies such as SignAI, and SLAIT attempting to create models that do live asl transcription. However, the products are still in development, and require cloud computing. In this project we propose models capable of live transcription, of the asl alphabet, with models that are efficient enough to be run live on consumer grade GPUs or even CPUs

Problem statement:



In this project we experimented with 3 different models. The 3 models are a ResNet Faster RCNN, a mobileNet Faster RCNN, and finally YOLOv5. The goal this project is to optimize these models for ASL translation and determine which model to use given task constraints.

Dataset

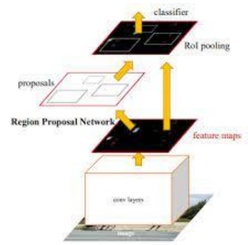


The full dataset contains 1512 images in the train set, 144 images in the validation set, and 216 images in the test set with a variety of backgrounds.Augmentations included, a horizontal flip does not affect ASL letters), a crop with 0-20% zoom, a rotation of±5, a vertical/horizontal shear of ±5, a grayscale to 10% of the images, a adjustment to brightness levels of±25%, and blur of up to 1.25 pixels. For Yolov5 additional augmentation were performed including a random affine, mosaics (pictures to the right) of different images, and a mixup of different images

Methods:

Faster RCNN Model:

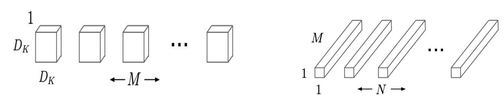
The Faster RCNN model is a two stage object detection model:
Region Proposal Network (RPN): Is the first stage of the model, utilizes convolutional neural networks to propose regions of interest.
Fast RCNN-detector: Uses the proposed regions of interest to make final detection decisions.



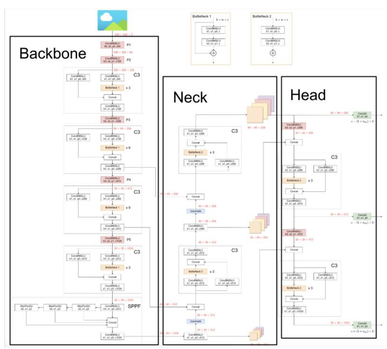
MobileNet:

The MobileNet architecture is a architecture of convolutional neural networks that is specifically tuned through a combination of hardware-aware network architecture tuning
to allow for optimal training on CPU architectures.
A unique architecture of the MobileNet is the replacement of convolutional layers with depthwise separable convolution which greatly reduces the computation necessary.
Depth Wise separable convolutions are composed of two parts:

- Filter convolutions: filters that are applied per input channel.
- Pointwise convolutions: 1x1 convolutions , used to create linear combinations of the depthwise layer.



Yolov5:



YOLOv5 like all models can be split into a backbone, neck, and head. The backbone is a New CSP-Darknet 53 which in composed of multiple CNN and C3, layers as well as a SPPF layer. The neck is made f many modified C3 blocks with upsampling. The head is very similar to the neck but it includes a final Conv2d layer at each depth to give the output.

Evaluation Metrics:

$$Recall = \frac{TP}{TP + FN} \qquad \text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})}$$

$$F1 = \frac{TP}{TP + 0.5(FP + FN)}$$

Detection:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

$$\text{mAP} = \frac{1}{|\text{classes}|} \sum_{c \in \text{classes}} \frac{|TP_c|}{|FP_c| + |FP_c|}$$

Experiments & Analysis

Task:

Models:
ResNet-50 FRCNN, MobileNet FRCNN, YOLOv5

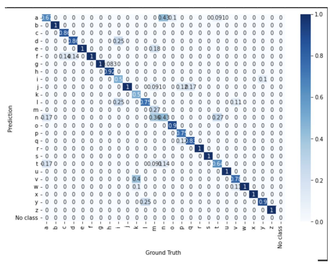
- Evaluation: F1 scores, Recall, Precision, mAP, inference speed, and parameter size
- Goal: contrast models through evaluating the models through the different evaluation metrics

Space/runtime eval	Parameters (in MB)	Mean CPU Inference Time(s)	Mean GPU Inference Time (s)
Faster RCNN ResNet-50	159.69	4.15	.1796
YOLOv5	27.9	.307	.0067
Faster RCNN mobileNet	72.89	.167	.0176

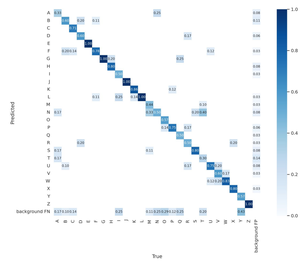
Model	Avrg Precision	Avrg Recall	F1 Score	mAP@0.5:0.95
Yolov5	0.65	0.737	0.717	0.501
ResNet FRCNN	0.84	0.83	0.82	0.735
MobileNet FRCNN	0.81	0.80	0.80	0.741

MobileNet FRCNN Confusion Matrix

ResNet FRCNN Confusion Matrix



YOLOv5 Confusion Matrix



Conclusions

Takeaways:

- We can see that each of the models that we test have their own merits: the ResNet FRCNN performs the best in classification tasks obvious choice if processing time in not an issue. however on the negative side, it is infeasible to run live on a CPU and has very low frame rate on a GPU and was by far the largest model in terms of parameters.
- The mobileNet FRCNN architecture was able to produce the best object detection metrics while maintaining extremely fast performance on the CPU, however it has more parameters than Yolov5 and performs worse in classification in comparison to the ResNet FRCNN.
- Yolov5 has much less parameters than the others and is lightning fast on a GPU.

Future Exploration

- , Expanding the dataset would help the model generalize better
- More expansive hyperparameter search should yield marginally better results.
- The alphabet is just the beginning, future work,should work with word corpuses.