Hand Wave

American Sign Language Recognition System

Mohamed Metwalli Noureldin

Aya Khames Khairy

Supervisor: Prof. Salah Selim

Department of Computer Engineering & Systems

Faculty of Engineering at Alexandria University

January 2023

Contents

Abstract

A real-time sign language translator is an important milestone in facilitating communication between the deaf community and the general public. We hereby present an approach of an American Sign Language (ASL) translator based on a convolutional neural network. We are going to utilize the pre-trained SSD model architecture for the real-time recognition of the ASL using the concept of transfer learning and to introduce an approach to deal with the dynamic gestures.

1 Introducion

1.1 Problem Statement

Communication between humans is a form of life, as we do this every single day and we may need to communicate with people who do not speak the same language as ours, but despite the many languages, communication has become easier because of translators.

Now, what about the deaf and dumb people, couldn't it be easier to communicate with them..? As a way to help facilitate this problem, we decided to make a software that detects one of the sign languages (ASL - American sign language) and converts the detected signs captured using a real-live camera to text with the ability to convert it to audio too.

We are going to use the SSD architecture by applying transfer learning to one of it's versions over the data we're going to collect for +40 classes.

Our software will detect ASL numbers, alphabets and some other of its known words and the number of words will be determined according to the suitable datasets found.

1.2 Assumbtions

We add dynamic letters of the American Sign Language. All the research works that we read about were excluding them because they were including motion, and we will deal with it by taking a frame for every possible pose of the motion of the letters which will increase the accuracy.

2 Related Work

ASL recognition is not a new computer vision problem, it has been implemented using the traditional convolutional neural networks even with the models built and trained from scratch or the use of the pre-trained models by applying transfer learning over a smaller dataset.

A Project [4] for the same problem but not real-time implemented it using transfer learning for the pre-trained VGG-16 model, and used self-collected data but with no variety, data splits "training, validation and testing" are almost the same, the data is collected by only one person with no noise in the background or different lighting conditions, or different views for the same letter.
Cropping the section of images that include the hand depends on the contrast, which might fail in the case of having noise or non-flat regions in the image, because of the similarity between the training and testing sets, they had good accuracies approximately 98%.

This paper [1] implemented the same problem in the real time using a website, they used the pre-trained GoogLeNet model, and used two different datasets each collected by different five persons with variety in the: skin color, poses of the letter gesture, lighting conditions and background.
Cropping the section of images that include the hand is done using zero padding and random cropping so they reduce the probability of losing a pixel which is related to the hand.
They tested the model in different ways, tested over many sets of letters (a-y, a-k, ..) with different parameters, on the data of the fifth person from each dataset, as the model trained only on the first four persons' data from each, and has been tested at the real-time too, the resulted accuracies were 70% - 98%.

3 Technical Approach

3.1 Model Architecture

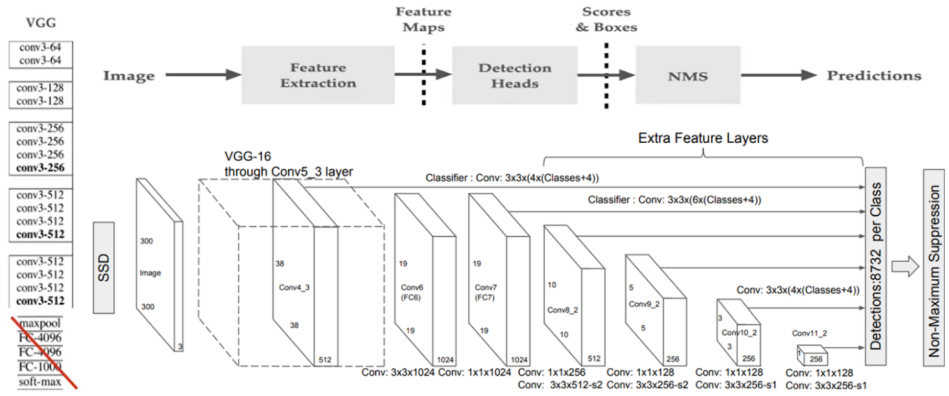

Figure 1: The basic SSD model architecture consists of a base convolutional neural network like VGG-16 which is truncated before the classification stage, followed by feature maps in different scales that decrease progressively in size, these steps are responsible for the feature extraction, followed by detection heads as small size conv. layers to either produce a score for a category or a shape offset relative to the default box coordinates, followed by a non-maximum suppression layer [3].

3.1.1 Base Network (Feature Extraction)

The VGG-16 network is used as a base, but other networks should also produce good results like ResNet, these earlier layers are based on a standard architecture used for high-quality image classification.
The VGG is used without the final stage of preparing for classification to get the resulting output as a good starting point for the feature maps.

3.1.2 Multi-Scale Feature Maps for Detection (Feature Extraction)

Convolutional feature layers are added to the end of the truncated base network, they decrease in size progressively and allow predictions of detections at multiple scales, the convolutional model for predicting detections is different for each feature layer.

3.1.3 Convolutional Predictors for Detection (Detection Heads)

Each added feature layer (or optionally an existing feature layer from the base network) of size (m*n*p) can produce a fixed set of detection predictions using a set of convolutional filters of size (3*3*p) these small kernels produce either a score for a category or a shape offset relative to the default box coordinates.

3.1.4 Non-Maximum Suppression (NMS)

In this stage, the multiple detections for the same object are eliminated to one detection.

3.1.5 Default Boxes and Aspect Ratios

A set of default bounding boxes (k) is associated with each feature map cell, for multiple feature maps at the top of the network, at each feature map cell, predict the offsets relative to the default box shapes (4) in the cell, as well as the per-class scores (c) that indicate the presence of a class instance in each of those boxes $\rightarrow$ k(4 + c) for each cell in the different feature maps scales.
Combining predictions from multiple feature maps with different resolutions to naturally handle objects of various sizes and allowing different default box shapes in several feature maps lets us efficiently discretize the space of possible output box shapes.

3.2 Data

Figure 2: shows samples of the self-collected dataset especially for the traditional CNN model, with different lighting conditions and backgrounds, cropped only around the hand.

We have collected data that we were going to use if we implemented the traditional CNN, but as it's shown in [1], it's very slow and needs much processing for the input frames before detection in the real-time like the case of the faster R-CNN [2] that include a region proposal stage, which makes it more slowly, as we decided to use the architecture of the SSD model, so we are going to collect the data in a different way to contain the whole person not only the hands and add the ground truth bounding boxes to it.
To get more data of the data we are going to collect, we will use the original image and its flipped version, as the hand gestures can be made with either the right or the left hand and this is a kind of the data augmentation technique to make the model more robust to these kind of transformations.
We are going to add the dynamic letters(j, z) and try to deal with them by

using different poses during the motion per each of them.

3.3 Transfer Learning

Models are usually trained over a huge datasets, and the training phase consumes much time and resources. This technique allows to retrain a prtrained model on a smaller amount of data than the data it was pretrained on.
The knowledge transfer is done by firstly, freezing the weights (hyperparameters) of the model, all of them or some of the earlier ones according to the size of the retraining data, secondly, the fully connected Layers in the original model is chopped off, and replaced with customized Fully-Connected Layers for the new dataset to deal with the new number of classes, finally any unfreezed parameters are going to be tuned during the training.
The primary benefits of such a technique are its less demanding time and data requirements.

3.4 Challenges and Approaches

The challenge in transfer learning comes from the differences between the original data used to train and the new data being classified, as the new data will be having high similarity between classes because they all are different poses for the same object in the old dataset which is the hand object.
we are going to try training only one model over the whole data (letters, numbers and some words) and try to get good detection results, if the results weren't good, we are going to train three models one for the letters, one for the numbers and one for the words to reduce the similarities by reducing the number of classes that the model needs to distinguish between.

4 Evaluation

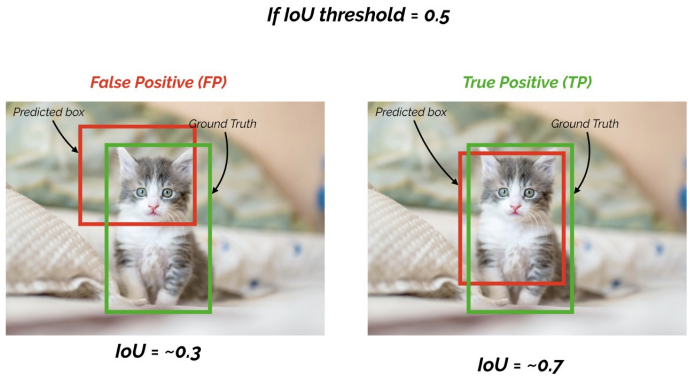

Figure 3: intersection over union (IoU) with 0.5 threshold to determine whether the detection is accepted or not

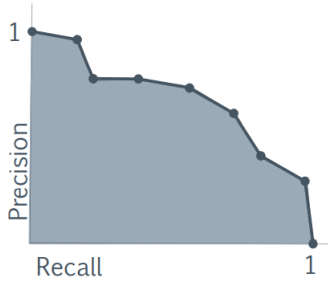

Figure 4: Precision - Recall curve, area under the curve presents the average precision (AP)

We are going to evaluate the model using the mean average precision (mAP), and the number of processed frames per second (FPS).
The mean average precision depends on the IoU technique that calculates the area of intersection between the ground truth box and the matching box divided by the union area of both boxes, then apply a threshold over it to decide if it's an accepted match or not to use these matches later to calculate the precision and the recall.
Precision is the number of correct matches over the total number of matches, while the recall is the number of the ground truth boxes with correct matches over the number of the ground truth boxes, so if we increased the threshold, we get higher recall and lower precision, and if we reduced it, we get lower recall and higher precision.
For each class the area under the precision-recall curve is called average precision AP, and mAP is the mean of the APs of the classes.

5 Conclusion

We are going to build a website application named "Hand Wave" as a real-time translator for the American sign language using the SSD architecture as it offers good detection results and multi-box detections with high speed compared to other architectures, we are going to collect the data on our own with adding the ground truth to them including the dynamic letters to try to deal with them by acting with the different letter poses in the same way. more data makes better so data augmentation techniques would help to get more data from the original one by applying some transformations on them.
The SSD model reduces the needed time for classification and also gets rid of the automatic hand detection and cropping problem.

6 Tools to Be Used

- Flask framework.
- Offline Python IDE like Jupyter NoteBook.
- Online Python IDE like Google Colab if needed due to GPU requiring.
- HTML, CSS, JavaScript, with VScode IDE.

7 Future Work

Additional work to be done is to add more languages, not only the American language, by retraining the model over the new data with our data too which is known as (transfer learning), or to make a model for each language to get better accuracies as it reduces the similarities between the classes, or to improve our model (architecture and parameters) to get better detection accuracy.

Appendix

Models Architectures Comparison

	Faster R-CNN	YOLOv3	SSD
Phases	RPN + Fast R-CNN detector	Concurrent bounding-box	regression and classification
Neural Network Type	Fully convolutional	Fully convolutional	Fully convolutional
Backbone Feature Extractor	VGG-16 or other feature extractors	Darknet-53 (53 convolutional layers)	VGG-16 or other feature extractors
Location Detection	Anchor-based	Anchor-Based	Prior boxes/Default boxes
Anchor Box	9 default boxes with different scales and aspect ratios	K-means from coco and VOC, 9 anchors boxes with different sizes	A fixed number of bounding boxes with different scales and aspect ratios in each feature map
IOU Thresholds	Two (at 0.3 and 0.7)	One (at 0.5)	One (at 0.5)
Loss Function	Softmax loss for classification; Smooth L1 for regression	Binary cross-entropy loss	Softmax loss for confidence; Smooth L1 Loss for localization

ASL Alphabets

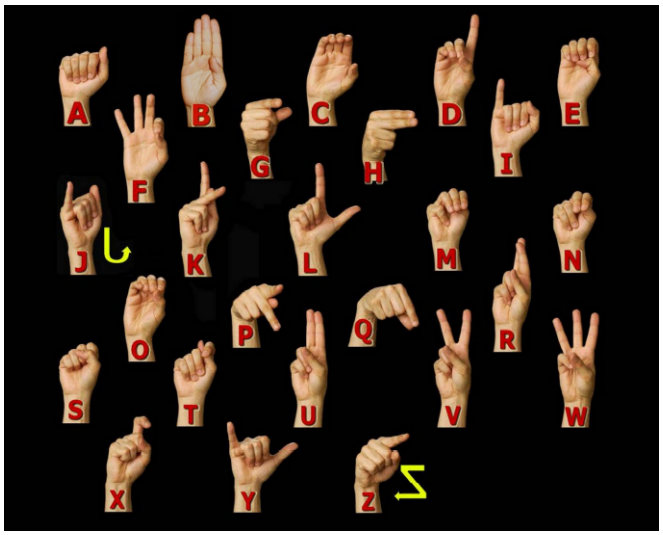

Figure 5: American sign language fingerspelling alphabet.

SSD Objective Function

The overall objective loss function is a weighted sum of the confidence loss L_{conf} and the localization loss L_{loc}:

$$L(x,\ c,\ l,\ g)\ =\ \frac{1}{N}(L_{\text{conf}}(x,\ c)\ +\alpha L_{\text{loc}}(x,\ l,\ g))$$

where N is the number of matched default boxes, α is set to 1 by cross validation. The confidence loss is the softmax loss over multiple classes' confidences(c).

$$L_{\text{conf}}(x,\ c)\ =\ -x_{\text{ij}}^{p} log(\hat{C}_i^p) - log(\hat{C}_i^0)$$

$$\hat{C}_i^p\ =\ \frac{e^{c_i^p}}{e^{c_i^p}}$$

Where $x_{\text{ij}}^{p} = \{0,\ 1\}$ is an indicator for matching the i-th default box to the j-th ground truth box of category p.

Localization loss is a Smooth L1 loss between the predicted box (l) and the ground truth box (g) parameters, we regress to offsets for the center (cx, cy) of the default bounding box (d) and for its width (w) and height (h).

$$L_{\text{loc}}(x,\ l,\ g)\ =\ (\ x_{\text{ij}}^{k}\ \text{smooth}_{L1}(l_i^m\ -\ \hat{g}_i^m))$$
$$\hat{g}_j^{\text{cx}}\ =\ \frac{g_j^{\text{cx}}\ -\ d_i^{\text{cx}}}{d_i^w}\ ,\ \hat{g}_j^{\text{cy}}\ =\ \frac{g_j^{\text{cy}}\ -\ d_i^{\text{cy}}}{d_i^h}\ ,\ \hat{g}_j^w\ =\ log(\frac{g_j^w}{d_i^w}),\ \hat{g}_j^h\ =\ log(\frac{g_j^h}{d_i^h})$$

Acknowledgements

We would like to thank Prof. Salah Selim for his precious guidance, Dr. Marwan Torki for his effort with us in the computer vision course, and Dr. Andrew Ng for his excellent content in the computer vision networks that is available on the internet for everyone and that was one of the main resources.

References

[1] Brandon Garcia and Sigberto Alarcon Viesca. Real-time american sign language recognition with convolutional neural networks. *Convolutional Neural Networks for Visual Recognition*, 2:225–232, 2016.

[2] Min Li, Zhijie Zhang, Liping Lei, Xiaofan Wang, and Xudong Guo. Agricultural greenhouses detection in high-resolution satellite images based on convolutional neural networks: Comparison of faster r-cnn, yolo v3 and ssd. *Sensors*, 20(17):4938, 2020.

[3] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[4] Zach Carlson Andrew Napolitano Tyler Beard, Adam Bennion. Asl image recognition. `https://github.com/zachcarlson/ASLImageRecognition`, 2022.