Machine Learning Engineer Nanodegree

Capstone Proposal

Victor Geislinger
2018 July 31

American Sign Language Handshape Recognition from Static Images

Domain Background

American Sign Language (ASL) is a sign language that does not use speech to communicate and is mostly used by the American Deaf population. Though used throughout the English-speaking United States, it is in fact its own language seperate from English and relies on building the language's syntax with multiple visuals such as handshapes, movements, positions, and nonmanual markers. Although there are many variations of sign language specific to different languages, regions, and needs, being able to use a computer to detect ASL would be extremely useful in not only ASL translation but also other sign language translations as well as non-language geusture recognition.

There has been past research in detecting ASL or ASL-like handshapes and movements such as gesture recognition. There have been past attempts in detecting hand motions and handshapes that use datasets with depth information using sensors like the Microsoft Kinnect. However, the technology required for this type of data is relatively uncommon compared to the ubiquitous camera sensor found on nearly every computer and phone. The recent advances in image recognition and classification make the concept of detecting and classifying ASL handshapes to be attainable through video or static images (possibly grabbed from video).

Problem Statement

This project will classify static images of different ASL handshapes. This is a good stepping stone before a video dataset is used in future projects/research since time dependence and movement will not have to be considered. My solution could be compared to classifying the MNIST handwritten database but with images of ASL handshapes. Deep learning could be used to classify the images. The model could then be evaluated with a validation set and/or new images from a similarly pre-proccessed but independent dataset.

Datasets and Inputs

The prefered dataset to be used will be the ASL FingerSpelling Dataset from the University of Surrey's Center for Vision, Speech and Signal Processing [1][2]. This dataset contains both colored images and depth sensing data collected from a Microsoft Kinnect. (Note that this project will not be using the depth sensing data since the project will be focused on using static images.) The images are in color and include 24 different handshapes each representing a letter from the English alphabet; "J" and "Z" are excluded since these letters are dependent on movement and therefore don't have a static image representation.

The images have been cropped around the handshape though each cropping results in a differently sized image fitting within a 200x200 window with a resolution of 72 pixels per inch. The background behind the handshape is not uniform or consistent. The dataset contains approximately 60,000 images generated by 5 different non-native ASL signers with over 500 samples of each of the 24 different handshapes. The handshapes in the image feature some rotational differences as the subject was instructed to adjust hand position for the camera.

Solution Statement

Deep learning will be used to classify the different handshape images, specifically using a convolutional neural network (CNN) as it has been historically been effective in many image recognition tasks. Transfer learning from previously trained models like VGG-16 can be used to help with the efficiency of this task. The data to be used already has been cropped with inconsistent sizes but around the handshape. Therefore the images will have to be resized for the model's CNN architecture to function. The data can be used for training and validation via a 80-20 split respectively. The results can be presented as a confusion matrix of the 24 different handshapes and the accuracy/precision can be calculated to gauge its performance.

Benchmark Model

The first and simplest benchmark will be comparing the developed model with a "random choice" model. With each handshape being equally likely, we would expect the "random choice" model to only identify handshape instances $\frac{1}{24} \approx 4.2\%$ of the time on average.

A goal of this project is to achieve better performance than if specialized equipment other than a camera to take images were to be used (like the Microsoft Kinnect). We can then use the performance of the *"Spelling It Out"* paper's random forrest model as our idealized benchmark. Observing the paper's confusion matrix which used four of the subjects to train and validate the model and one subject's images to test performance, similar handshapes were misclassified with the handshapes least correctly identified being "T", "O", "S", and "M" (7%, 13%, 17%, and 17% of the time correctly identified respectively). The handshapes that were identified most accurately were "L", "V", "B", and "G" (87%, 87%, 83%, and 80% of the time correctly identified respectively). The paper's model achieved an overall mean precision of 73% and 75% using only the images and using both images and depth data respectively.

Evaluation Metrics

When comparing our model to the random benchmark model, we can evaluate the performance by looking at the number of correct handshapes it classfies $accuracy = \frac{N_{correct}}{N_{predicted}}$ where $N_{correct}$ is the number of correct predictions and $N_{predicted}$ is the total number of predictions made. When comparing the model with the *"Spelling It Out"* paper's model, we can get more complex. To represent the precision and recall of the model, we can calculate the F_1 score given by $F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$. We use a F_1 score over a different F_β ($\beta \neq 1$) score since both precision and accuracy should be considered with approximately the same weight (we care about classifying the correct hanshape as much as we care about misclassifying a letter).

Note we will also use categorical (multinomial) cross-entropy loss that penalizes predictions that are confident but incorrect to measure the performance of the model. This is given by

$$\sum_{l}^{M} y_{o,l} \cdot log(p_{o,l})$$

where l is the handshape class, M is the number of different handshapes, $y_{o,l}$ is either 0 or 1 indicating if the handshape label l is the correct classification for observation o, and $p_{o,l}$ is the predicted probability observation o is the handshape label l. The smaller the value, the better the performance of the model.

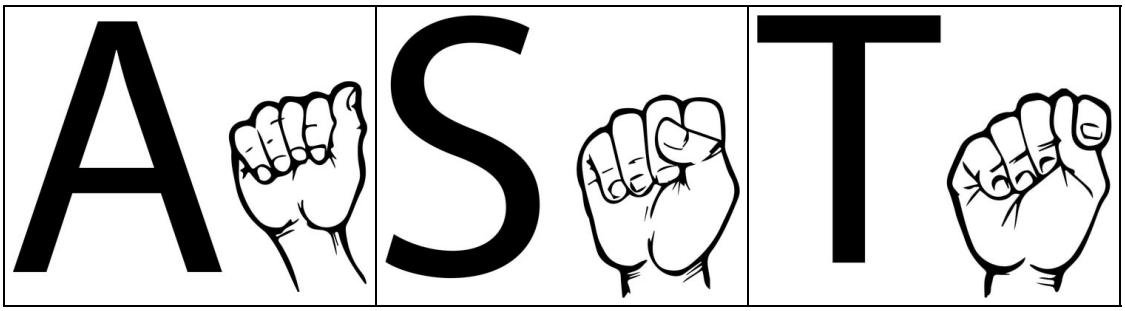
Project Design

The first step in this project will be to handle the dataset that will be used to train, validate, and test the model developed. The dataset contains both color (intensity) images and depth sensing data. Since our model will be generalized so it doesn't use any specialized sensor equipment, these depth sensing files will be removed. Next the images should be renamed so that it is clear what handshape and which of the subjects made the handshape (in the format of `handshape_subject_numericalid.png`). This will allow us to keep the data in one directory while still being able to identify and sort by the metadata of the images. This is done because one of the subjects' images will be kept as a seperate testing set to mirror the *"Spelling It Out"* paper's methodology.

Continuing with the dataset, the next step will be to do any preprocessing. The images will be grayscaled to reduce file size. This will also be done since color should not be relevant in classification. This step could be left out since dimension reduction will essentially remove this extraneous information but doing this "removal of color" will also generalize the model so that other datasets could be used in the future. The images will also need to be resized to the same dimension (square) which will be used in the CNN model. To determine the images' new size, we will do a quick exploration to see how the images compare amongst each other.

After all this preprocessing is completed, the dataset will be split so that one subject's images will be reserved for testing while the other four subects' images will be split so that 80% of this set will be designated for training and the remaining 20% of the set will be designated for validation. Another independent dataset may also be used to test the performance. This independent dataset could be used to confirm the performance on the 5th subject's images.

After preprocessing and splitting the data, the model can be built. The model will be close in structure with what was used in the MLND dog identification project. The model (made from scratch) will likely need multiple CNN layers with dropout and pooling as well as fully connected layers with outputs for each of the 24 different handshapes. Here hyperparameters like stride length can be adjusted to balance efficiency with information extraction via the model's complexity. The biggest concern is to ensure the model picks up on the relative position of the fingers in the handshape since many handshapes have similar finger positions (see handshape images "A", "S", and "T" below). To improve performance, transfer learning from a publicly available pre-trained network such as VGG-16 can be used. This will however depend on the available computing resources for this project.



Similarities amongst three different handshapes (images provided by www.supercoloring.com).

After the model has been trained, its performance can be evaluated by making predictions on the testing dataset. The most straightforward metric for evaluation is to simply calculate the accuracy of the model's predictions (percentage of handshapes correctly predicted). This can be compared with the "random model" which would theoretically predict handshape instances correctly approximately 4.2% of the time. The model's

performance can also be measured by looking at the categorical cross-entropy loss which will produce a value that represents not only how correct the model's predictions are but also on how confident it was on those predictions.

After that initial evaluation, the model will be compared with the *"Spelling It Out"* paper's model. This will be done by producing a confusion matrix for the different handshapes our model predicts and then comparing the confusion matrix from the paper (see below). From these predictions, a F_1 score will be calculated which will provide a numerical metric to directly compare with.

	a	b	c	d	e	f	g	h	i	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y
a	**0.75**		0.05									0.05		0.05					0.10					
b	0.03	**0.83**	0.03								0.03								0.07					
c			**0.57**	0.13	0.03	0.03	0.03		0.07		0.03								0.03					0.03
d				**0.37**		0.13	0.03		0.07	0.03	0.07						0.17	0.03					0.10	
e		0.07	0.03		**0.63**								0.03	0.03			0.03	0.10	0.07					
f		0.30	0.10		0.05	**0.35**					0.15							0.05						
g	0.05				0.05	0.05	**0.60**								0.20				0.05					
h							0.03	**0.80**		0.03			0.03		0.10									
i		0.03	0.03	0.03		0.03			**0.73**				0.03						0.03				0.03	
k			0.03	0.03		0.07	0.03			**0.43**	0.03		0.03				0.07				0.20		0.03	0.03
l				0.13							**0.87**													
m	0.10		0.03		0.10		0.03				0.03	**0.17**	0.10		0.03	0.03		0.27					0.07	
n	0.17	0.10		0.03					0.03			0.10	**0.23**	0.07				0.13	0.10		0.03			
o	0.10		0.30	0.13		0.03	0.07		0.03	0.07	0.03			**0.13**	0.07			0.03						
p			0.07	0.10		0.03		0.10	0.03						**0.57**	0.07			0.03					
q	0.03						0.07								0.07	**0.77**			0.03				0.03	
r			0.03	0.03	0.03	0.07			0.03								**0.63**			0.13	0.03			
s	0.30		0.13	0.03	0.07				0.13				0.03	0.03				**0.17**	0.07				0.03	
t	0.33				0.13			0.03	0.03			0.07	0.03	0.10				0.20	**0.07**					
u		0.17		0.03													0.10			**0.67**		0.03		
v			0.03								0.03						0.03			0.03	**0.87**			
w			0.03			0.03								0.03							0.37	**0.53**		
x	0.03	0.03		0.17		0.07	0.07		0.20	0.03			0.07				0.10						**0.20**	0.03
y	0.07					0.07			0.10															**0.77**

*Confusion matrix of predictions for model in **"Spelling It Out"** paper.*

Finally, an independent dataset of handshapes could be used to demonstrate the robustness of the model. This could be a similarly sized dataset of cropped images or simply images taken of new subjects making the different handshapes. This dataset would be less about testing the model's performance and more about the application of the model for other uses beyond the project and its goals.

References

[1]: Pugeault, N., and Bowden, R. (2011). Spelling It Out: Real-Time ASL Fingerspelling Recognition In Proceedings of the 1st IEEE Workshop on Consumer Depth Cameras for Computer Vision, jointly with ICCV'2011

[2]: Pugeault, Nicolas. "Nico" ASL FingerSpelling Dataset from the University of Surrey's Center for Vision, Speech and Signal Processing, empslocal.ex.ac.uk/people/staff/np331/index.php?section=FingerSpellingDataset.