ASL Alphabet Recognition Project

Federico Alvetreti 1846936, Aurora Bassani 1852791, Ioan Corrias 2079420,
Erica Luciani 1868647, Gabriele Pelliccioni 1838084

Abstract

The following report contains a solution to an image classification task about American Sign Language alphabet, using a Convolutional Neural Network.

1 Introduction

The main idea behind our project was to make sign language accessible to more people, to promote social inclusion of signing people. In order to do so, we decided to start with a simple task, namely classifying ASL alphabet images.

2 Proposed method explained

Initial idea At first we tried feeding a fully-connected net with pre-processed images: we resized the images, converted them in black and white and then we used different filters (gradient descent, sharpening and hysteresis) to detect the edges. However some images had a lot of noise in the background and so the net wasn't able to discern the hands from some parts of the background, resulting with a low accuracy.

CNN Although fully-connected networks make no assumptions about the input, they tend to perform less and aren't good for feature extraction, hence we decided to change approach and use a Convolutional Neural Network (CNN). CNNs are a subset of neural networks, particularly useful for finding patterns in images. Every CNN is composed by the following:

- **convolutional layers**: main components of the CNN, used to extract features from the images;
- **pooling layers**: decrease the dimension of the image, reducing also the quantity of parameters and computations done by the net;
- **fully-connected layers**: flatten the output from the previous two layers and use it to get the probability for each class of images.

We used a validation set to implement the early stopping regularization technique which avoid overfitting the model. Our logic was to stop the training phase after the validation loss doesn't improve for two epochs.

3 Dataset description

At first we found one dataset from Kaggle with 29 classes (one for each letter sign and three for the signs for 'delete', 'nothing, 'space') and 3000 images per class. The images had different level of brightness ad different zoom level -sometime it was only the hand, sometime the wrist was shown and sometimes even part of the arm- but we noticed that all the images had a similar simple background, hence we decided to add a second dataset from Kaggle containing only 30 images per class but with a lot of noise in the background (buildings, balustrades...). We merged the two merged datasets, but of course we had to make the final dataset balanced, in terms of images type and number of images. We selected 140 images per class, taking all the images in the second dataset and choosing images from the first paying attention to pick the same amount of images for each variety.

4 Experimental results

We tried different learning rates, in the range [0.1, 0.000001]. Too high rates will make the learning jump over minima while too low ones will either take too long to converge or get stuck in an undesirable local minimum.
To balance this trade-off we recognized a learning rate of 0.00001 as the best for our model. The overall accuracy of our model is 76,6%. Since this type of value doesn't show us effectively how the net perform for each class we've also computed a table that shows the True Positive Rate for each class and also the other two classes that the model identifies instead of the correct class.

	Predicted	Second Choice	Third Choice
A	A : 0.83	B : 0.06	V : 0.06
B	B : 0.94	Q : 0.06	
C	C : 0.67	I : 0.17	B : 0.11
D	D : 0.89	F : 0.06	T : 0.06
E	E : 0.78	A : 0.06	B : 0.06
F	F : 0.67	E : 0.17	C : 0.06
G	G : 0.89	P : 0.11	
H	H : 0.72	D : 0.17	J : 0.06
I	I : 0.83	H : 0.06	J : 0.06
J	J : 0.83	M : 0.06	V : 0.06
K	K : 0.89	H : 0.06	S : 0.06
L	L : 0.83	Y : 0.11	X : 0.06
M	M : 0.83	K : 0.11	B : 0.06
N	N : 0.78	U : 0.17	R : 0.06
O	O : 0.78	L : 0.11	R : 0.06
P	P : 0.56	space : 0.17	Q : 0.17
Q	Q : 0.5	U : 0.17	nothing : 0.17
R	R : 0.5	U : 0.17	V : 0.17
S	S : 0.5	X : 0.17	Z : 0.17
T	T : 0.89	S : 0.06	space : 0.06
U	U : 0.78	N : 0.11	Y : 0.11
V	V : 0.89	E : 0.06	N : 0.06
W	W : 0.39	V : 0.17	X : 0.17
X	X : 0.83	A : 0.06	Y : 0.06
Y	Y : 0.78	S : 0.17	L : 0.06
Z	Z : 0.94	H : 0.06	
del	del : 0.67	X : 0.17	Y : 0.11
nothing	nothing : 0.94	R : 0.06	
space	space : 0.89	Y : 0.06	nothing : 0.06

5 Conclusions and Future work

Despite we tried to create a balanced dataset, we believe that, to improve our results we should increment the number of images for class and also the variety of them (for example taking pictures from people of different gender and age).
Another way to improve the model could be cropping the images to get only the hand part out of them, this would solve the problem of noisy backgrounds which takes down the performance of our net. This could be done using a specific neural model that performs an image identification task which output only the part of the picture with the hand.

6 References

- https://www.kaggle.com/datasets/grassknoted/asl-alphabet
- https://www.kaggle.com/datasets/danrasband/asl-alphabet-test
- https://jovian.ai/aakashns/05-cifar10-cnn
- https://www.kaggle.com/code/swamita/asl-classification-using-cnns-keras-99-89-acc

7 Members' roles

We had the chance to collaborate the whole time, hence we didn't divide the work, instead everyone worked actively during each step of the project.