American Sign Language Recognition with Neural Networks

Esteban Galvis
Universidad de los Andes
Cra. 1 Este Nº 19A - 40
e.galvis10@uniandes.edu.co

Abstract

The following paper discusses traditional computer vision recognition techniques with state-of-the-art deep learning approaches. The task in place is to recognize 24 letters of the alphabet of ASL (American Sign Language).The following methods were analyzed: SIFT feature extractor with a Suppor Vector Machine classifier, AlexNet's training architecture and transfer learning with ResNet-18.

1. Introduction

One of the main problems in `Computer Vision is Image Recognition. The problem is to be able to segment and classify a region of an image (or the image) to a specific category. Since traditional Template Matching [15], image recognition methods have evolved to using Neural Networks.

The goal of this paper is to be able to recognize a hand gesture that is conveying a language that is unknown for a clear majority of people: American Sign Language (ASL). Having a numerous amount of people that can't understand American Sign Language means that the deaf community is completely isolated from a large portion of society. This is unacceptable and must be issued.

This task is therefore an image recognition problem applied to finger and hand gestures. As it was mentioned this is an important issue that must be worked on, and thanks to today's research, this type of task can be achieved and have important and positive results in society. The idea is to be able to classify a pair of finger and hand gesture images with their corresponding meaning in ASL. Thus, this is a joint problem of image segmentation and recognition.

To be successful in this challenge, a supervised approach is going to be applied. Deep learning, which is a machine learning subfield that stacks computational neural networks to learn representations of concepts [1] is going to be used to classify such images. This artificial intelligence technique has achieved great results in question-answering [2], image captioning [3], machine translation [4][5], games [6] and image segmentation [7] tasks to name a few. The milestone that this paper wants to achieve is finger and hand gesture recognition.

2. Related Work

There are several techniques to classify images that have been developed through time. Some are more related to traditional computer vision algorithms, and others are more inclined to deep learning.

2.1. Traditional techniques

Traditional work done to solve this task is segmenting an image with a color skin model to detect if there are hands present in the image over a Markov Random Field (MRF), while using Histograms of Oriented Gradients (HOG) as a hand shape descriptor and a Hidden Markov Model (HMM) classifier to combine a lexicon of known words [8]. The proposed method applies to the British Sign Language but its implementation differs with this paper's baseline classifier since it is a Support Vector Machines and they use Markov Models. Yet, both calculate HOG in their implementation.

Another method, uses a Gaussian Mixture Model and color histogram cues to segment according to a skin model. After these have been applied, a depth based segmentation is done thanks to Time-of-Flight (ToF) cameras. Classification of the gestures are done using Haarlet coefficients that are trained on an Average Neighborhood Margin Maximization algorithm. [9] This method has a disadvantage of using cameras that make data more prone to variation, while our method is invariant. Additionally, there has been work using a Microsoft Kinect to get depth images and extract hand shape features using a Gabor Filters with different scales and orientations with intensity and depth cues. The output is then classified using a multi-class random forest classification. [13]

The main difference with all these methods from the proposed one is that all the feature extraction is done by a neural network.

2.2. Deep learning techniques

On the other hand, recent work around the subject is usually based on neural networks. Convolutional Neural Networks (CNN) have been used to extract representation of features of an image. There has been work done using CNN additional to Max Pooling regularization in Hand-Based Recognition of 6 gestures [10]. This method is different from ours since we are classifying 24 classes instead of 6. Another approach is to create a feature intensity and depth vector and then feed it to a Deep Belief Network made of three Restricted Boltzmann Machines (RBM) to classify hand gestures with their corresponding alphabet letters [11].

Another worked used a CNN to recognize fingerspelling from a depth image, where hand segmentation was done using a black wrist band to create a depth void around the hand, making it easier to segment [12], we don't use such methods to help the localization of a hand.

3. Method

In overall, the field of computer vision is showing promising results with Neural Networks. To show this in the current ASL Fingerspelling recognition task, the following paper will extract SIFT[15] descriptors for each image and then create a visual vocabulary of our dataset to classify each one of them with a Support Vector Machine. This approach relates to traditional Computer Vision techniques and its results will be compare to new deep learning techniques to see which approach performs better. To solve this task using neural networks, AlexNet [16] will be trained with the dataset, and transfer learning strategies will be applied to fined-tuned ResNet-19 [17] with our dataset.

4. Dataset

The ASL Finger Spelling [13] dataset is going to be used. This set is composed of 24 classes that correspond to a letter of the alphabet. Five different individuals were recorded conveying one of the different 24 letters in ASL. See figure 0 for an example. The evaluation metric will consist of a Confusion Matrix for the image classification problem.

Figure 0.1: Example of a hand gesture of the letter B in ASL

4.1. Preprocess

Examining the dataset, I found that the minimum number of pictures per class was 2615, thus I prepared the dataset to have an uniform distribution of 2615 pictures per letter. The whole set is of 62760 pictures, where the training set and test set are of 43944 and 18816 respectively. In other words, 70% for training and 30% for testing.

5. Experiments

5.1. Baseline

For our baseline, (BOVW) Bag of Visual Words is going to be used to classify an image. For this approach, OpenCV[18] was used to calculate SIFT descriptors for each image. This makes the descriptor invariant to scale and rotations which makes it proper for solving recognition tasks. See figure 1 for an example of SIFT descriptors.

Figure 1: Example of the keypoint descriptors of the letter A

An important concept in the Bag of Visual Words model is the vocabulary, which contains all the 'visual words'. Since images per se are not words, a patch of the image will be an abstraction of a word. The baseline configuration is of 100 'words. A SVM was trained with it

with C=4. Please check figure 0 to see the results of the model.

	precision	f1-score	support	support
a	0.36	0.28	0.31	784
b	0.39	0.63	0.48	784
c	0.34	0.4	0.37	784
d	0.38	0.52	0.44	784
e	0.36	0.2	0.26	784
f	0.39	0.46	0.42	784
g	0.41	0.48	0.44	784
h	0.47	0.81	0.59	784
i	0.41	0.21	0.27	784
k	0.42	0.24	0.31	784
l	0.44	0.35	0.39	784
m	0.34	0.36	0.35	784
n	0.39	0.18	0.25	784
o	0.37	0.35	0.36	784
p	0.41	0.54	0.46	784
q	0.36	0.36	0.36	784
r	0.32	0.15	0.2	784
s	0.4	0.48	0.44	784
t	0.37	0.44	0.4	784
u	0.32	0.25	0.28	784
v	0.33	0.3	0.32	784
w	0.4	0.6	0.48	784
x	0.36	0.4	0.38	784
y	0.44	0.27	0.33	784
avg / total	0.38	0.39	0.37	18816

Figure 0: Results for a SVM with C=4 and a Vocabulary of 100

5.2. Neural Networks

Using a pre-trained ResNet19 network in ImageNet [19] , the last layer of the network was removed to add the categories of our fingerspelling dataset. This technique is known as transfer learning and it takes the image features learned from ImageNet to classify our current dataset. To be more exact, this approach takes great advantage of the original trained network's first layers and only trains the last layer to match its domain output. The PyTorch tutorial for this technique was implemented, which augments and transforms the data so its like the original data which it was trained with see Figure 2.

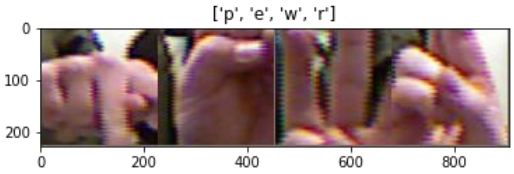


Figure 2: Data augmentation with labels

To fine-tune the network, batches of 4 images, a Cross-Entropy Loss Function was optimized using Stochastic Gradient Descent, using an initial learning rate of 0.001, a learning schedule that changed the learning rate 0.1 every 7 steps, and a momentum of 0.9. 10 epochs were run for a total of time of 328m 51s. See figure 4 and figure 5 to see the model's losses and figure 6 and figure 7 for the results.

Figure 3: Example of ResNet-18 output

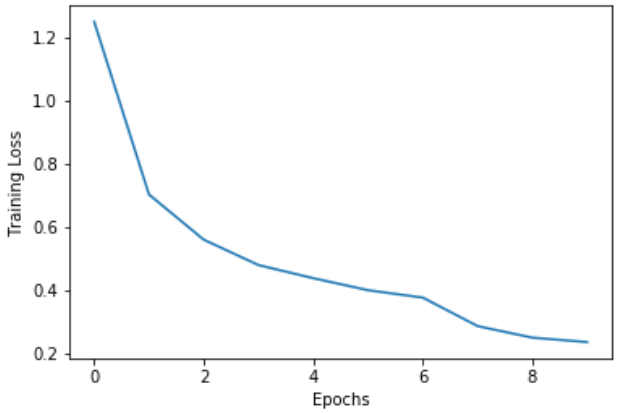


Figure 4: Training ResNet-18 fine-tuned loss

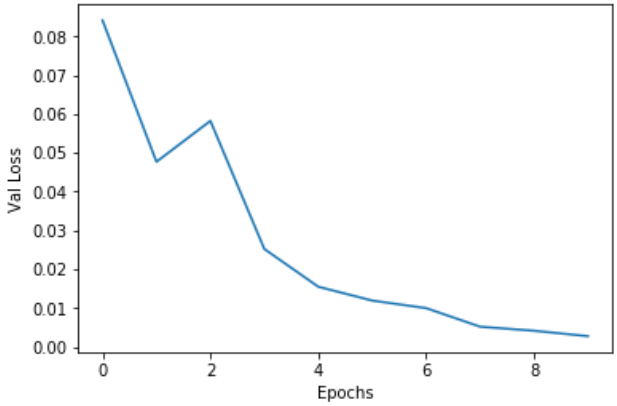


Figure 5: Test ResNet-18 fine-tuned loss

	precision	recall	f1-score	support
a	1	1	1	784
b	1	1	1	784
c	1	1	1	784
d	1	1	1	784
e	1	1	1	784
f	1	1	1	784
g	1	1	1	784
h	1	1	1	784
i	1	1	1	784
k	1	1	1	784
l	1	1	1	784
m	1	1	1	784
n	1	1	1	784
o	1	1	1	784
p	1	1	1	784
q	1	1	1	784
r	1	1	1	784
s	1	1	1	784
t	1	1	1	784
u	1	1	1	784
v	0.99	1	1	784
w	1	1	1	784
x	1	1	1	784
y	1	1	1	784
avg / total	1	1	1	18816

Figure 6: ResNet-18 test classification report

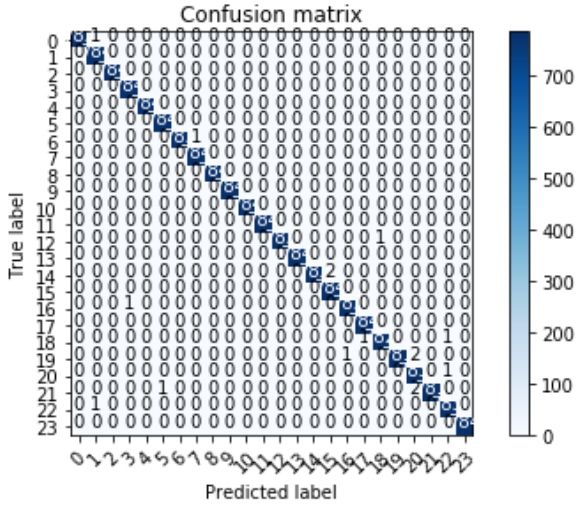


Figure 7: ResNet-18 fine-tuned test classification matrix

Nonetheless, there's another transfer learning strategy where the gradients are not fine-tuned with a dataset, but instead, freezes all the weights except the last layer and trains it so the network can act as a feature extractor. The same parameters mentioned before were used to train but it took only 131m 19s to train. See figure 8 and figure 9 to see the training losses. To see the results of the model please see figure 14

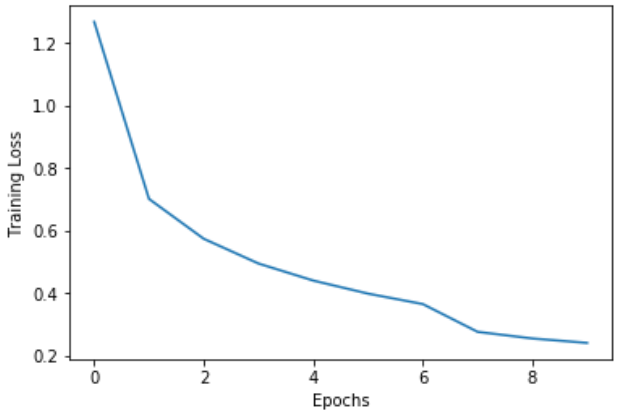


Figure 8: ResNet-18 feature extraction train loss

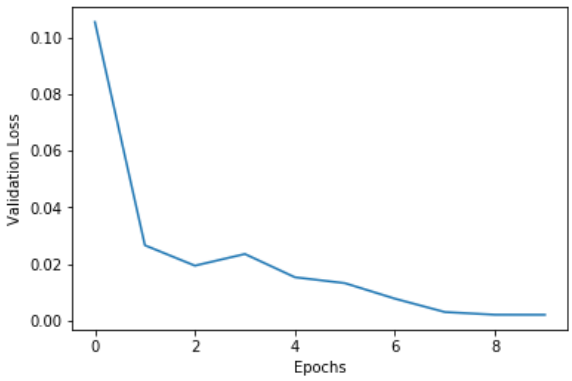
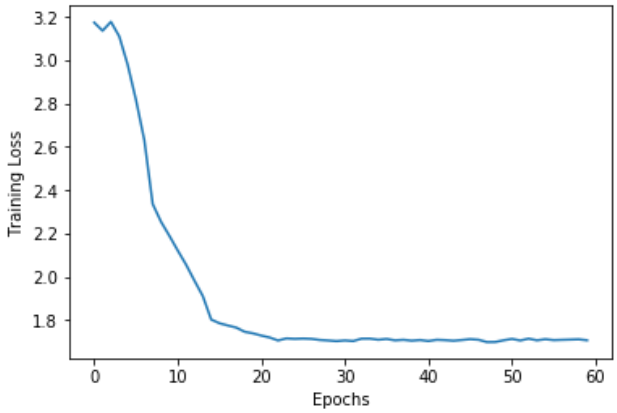


Figure 9: ResNet-18 feature extraction test loss

To analyze this method better, AlexNet's architecture was trained with our dataset for 399m 15s with 32 size batches. A Cross-Entropy loss was optimized with Stochastic Gradient Descent for 60 epochs with an initial learning rate of 0.003 and a scheduler that changed the rate by a factor 0.1 every 7 epochs, momentum of 0.9 and a weight decay of 0.0005. See figure 10 and figure 11 to see the model's losses. To grasp the model's results, see figure 12 and figure 13.



F Figure 10: AlexNet training loss

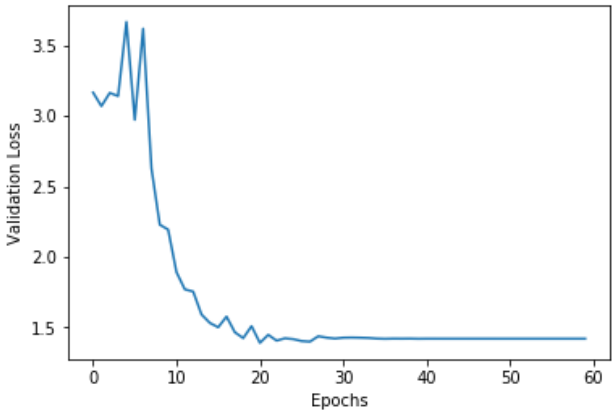


Figure 11: AlexNet test loss

	Train	Test	Training Time (hours)
ResNet-18 Fine-tuned	0.947365	0.99915	328m 51s
ResNet-18 Feature Extractor	0.945999	0.999415	131m 19s
AlexNet	0.453987	0.511586	399m 15s

Figure 14: Summary of results

	precision	recall	f1-score	support
a	0.64	0.82	0.72	784
b	0.58	0.84	0.68	784
c	0.47	0.84	0.6	784
d	0.63	0.31	0.42	784
e	0.32	0.46	0.38	784
f	0.36	0.59	0.44	784
g	0.6	0.8	0.68	784
h	0.83	0.48	0.61	784
i	0.55	0.3	0.39	784
k	0.64	0.4	0.5	784
l	0.52	0.98	0.68	784
m	0.47	0.29	0.36	784
n	0.37	0.32	0.34	784
o	0.69	0.53	0.6	784
p	0.59	0.65	0.62	784
q	0.78	0.46	0.58	784
r	0.72	0.07	0.13	784
s	0.36	0.33	0.35	784
t	0.38	0.3	0.34	784
u	0.44	0.39	0.41	784
v	0.92	0.3	0.45	784
w	0.95	0.33	0.48	784
x	0.35	0.54	0.43	784
y	0.46	0.94	0.62	784
avg / total	0.57	0.51	0.49	18816

Figure 12: AlexNet test classification report

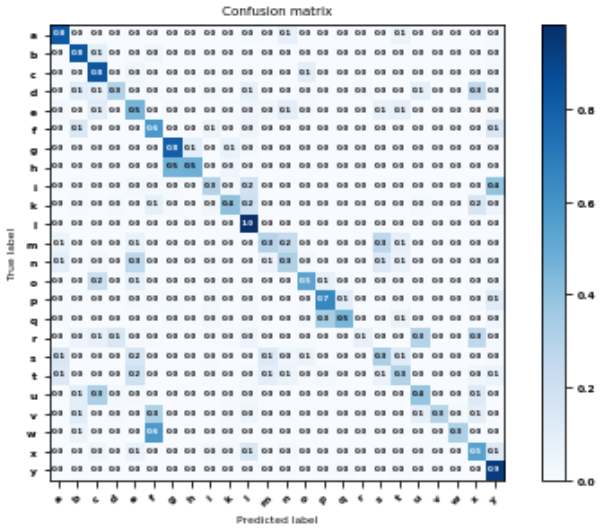


Figure 13: AlexNet confussion matrix

6. Conclusion

After all this research, is evident that the best option to choose for to recognize the alphabet of ASL is to use transfer learning on a ResNet-18 pre trained model, but not fined tuned it. It had the best results between all the methods and it was the fastest as

well. Yet, it is important to note that this method doesn't segment the image first to obtain the hand, but process all the image and manages to classify with accuracy. The implementation of AlexNet can be better if batch normalization is used and defined initialization method for the weights is introduced in order to train longer the network and obtain better results and not suffer for instance with vanishing gradients. This is not the case with the ResNet architecture since it can train with bigger networks since it doesn't have the vanishing gradient problem so much due to its residual property between layers.

References

[1] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. 2016. MIT Press

[2] Minghao Hu, Yuxing Peng, and Xipeng Qiu. Reinforced mnemonic reader for machine comprehension. CoRR, abs/1705.02798, 2017.

[3] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. CVPR, 2015

[4] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. CoRR, abs/1409.0473, 2014

[5] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning .

[6] Silver, David, Huang, Aja, Maddison, Chris J, Guez, Arthur, Sifre, Laurent, Van Den Driessche, George, Schrittwieser, Julian, Antonoglou, Ioannis, Panneershelvam, Veda, Lanctot, Marc, et al. Mastering the game of go with deep neural networks and tree search. Nature, 529(7587): 484–489, 2016

[7] K. Maninis, J. Pont-Tuset, P. Arbelaez, and L. V. Gool, "Convolutional ´ oriented boundaries: From image segmentation to high-level tasks," IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2017.

[8] S. Liwicki and M. Everingham. Automatic recognition of fingerspelled words in British sign language. In Proc. of CVPR, pages 50-57, 2009.

[9] M. Van den Bergh and L. Van Gool, "Combining RGB and ToF cameras for real-time 3D hand gesture interaction," in Workshop on Applications of Computer Vision (WACV), pp. 66-72, 2011.

[10] J. Nagi, F. Ducatelle, G. A. Di Caro, D. Ciresan,U. Meier, A. Giusti, F. Nagi, et al., "Max-poolingconvolutional neural networks for vision-based handgesture recognition," in Proc. of the 2nd IEEE InternationalConference on Signal and Image ProcessingApplications (ICSIPA), 2011, pp. 342-347.

[11]] L. Rioux-Maldague and P. Giguere, "Sign language fingerspelling classification from depth and color images using a deep belief network," in Computer and Robot Vision (CRV), 2014 Canadian Conference on, pp. 92–97, IEEE, 2014

[12] B. Kang, S. Tripathi, T. Nguyen, "Real-time sign language fingerspelling recognition using convolutional neural networks from depth map", Pattern Recognition 2015 3rd IAPR Asian Conference on, Nov. 2015.

[13] N. Pugeault, R. Bowden, "Spelling it out: Real-time ASL fingerspelling recognition", Proc. IEEE Intl. Conf. Comput. Vis. Workshops, pp. 1114-1119, 2011.

[14] V. Athitsos, C. Neidle, S. Sclaroff, J. Nash, A. Stefan, Q. Yuan and A. Thangali, *The ASL Lexicon Video Dataset*, CVPR 2008 Workshop on Human Communicative Behaviour Analysis (CVPR4HB'08)

[15] Lowe, D.G. International Journal of Computer Vision (2004) 60: 91. https://doi.org/10.1023/B:VISI.0000029664.99615.94

[16] Imagenet classification with deep convolutional neural networks A Krizhevsky, I Sutskever, GE Hinton - Advances in neural information processing systems, 2012

[17] Deep residual learning for image recognition K He, X Zhang, S Ren, J Sun - Proceedings of the IEEE conference on computer …, 2016

[18] G. Bradski. The opencv library. Dr. Dobb's Journal of Software Tools, 2000. 5, 6

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database. IEEE Computer Vision and Pattern Recognition (CVPR), 2009.