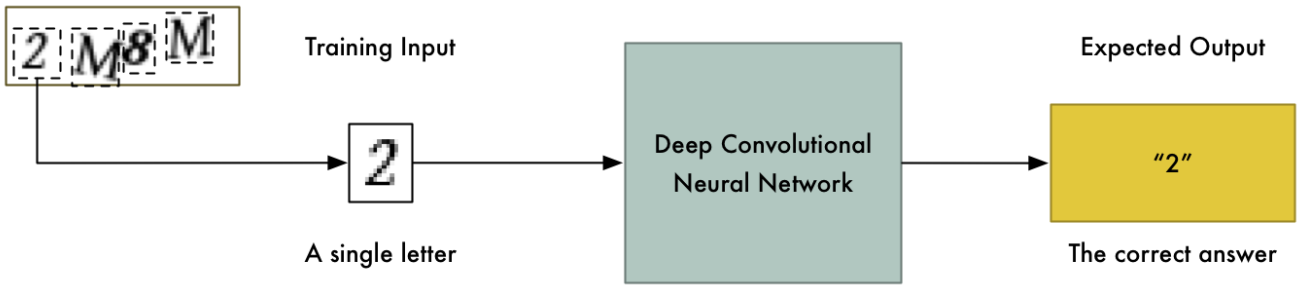
# Final Project. CAPTCHA Image Recognition using CNN and LSTM

Sangjun Son
Seoul National University
Department of Computer Science and Engineering
lucetre@snu.ac.kr

## 1. Introduction

Everyone hates CAPTCHAs — those annoying images that contain text you have to type in before you can access a website [1]. A CAPTCHA (an acronym for "Completely Automated Public Turing test to tell Computers and Humans Apart") is a type of challenge–response test used in computing to determine whether or not the user is human [2]. It is to prevent computers from automatically filling out forms by verifying that you are not a robot. But with the rise of deep learning and computer vision, decoding CAPTCHA seems to be possible and some of those systems are even more better than human.

Given an image, how can we recognize characters from the CAPTCHA image? Since each CAPTCHA image has a variable length of characters, considering length will be a significant work while training through deep neural networks. If we can split the image apart so that that each letter is a separate image, remaining work is just to train the neural network to recognize a single letter at a time[1].



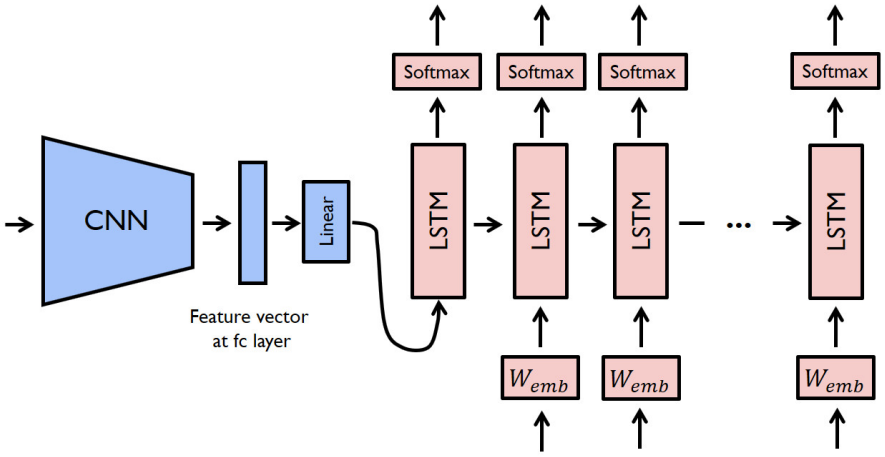

We've trained our network with 10,000 training examples whose lengths are in $2 \sim 5$ and tested with 1,000 test images of length $2 \sim 7$. Due to several parameters consisting the ensemble model of CNN and LSTM, we've conduct various experiments. After several experiments, we've found the best set of hyper-parameters that make our model perform more accurately and even faster.

## 2. Related Works

CATCHA recognition is one part of temporal pattern recognition projects. Simiarly, there are numerous examples like image captioning, speech recognition, part-of-speech tagging. Using ResNet and LSTM modules, which are the representatives of CNN and RNN methods, we can extract useful features from an image and continuously learning them over time.

## 3. Proposed Method

This section describes overall flows of our proposed method, `BetterNet+LSTM`. Feature extraction and learning the temporal patterns are the key concepts of our model.




### 3.1. Feature extraction by BetterNet

Since image vectors contain lots of information, we have to extract meaningful features. Pre-training inputs through ResNet, we

### 3.2. Learning temporal patterns by LSTM

The pretrained CNN extracts the feature vector from a given input image. The feature vector is linearly transformed to have the same dimension as the input dimension of the LSTM network. Using these source and target sequences and the feature vector, the LSTM decoder is trained as a language model conditioned on the feature vector [3].

## 4. Empirical Analysis

Here we describes how we came up with `jerk8vel8iter50` model. And this is the table of comparison in 1-iteration trained models in aspect of test accuracy.

| Model Name | jerk $j$ | acc $a$ | vel $\alpha$ | Accuracy |
|---|---|---|---|---|
| `jerk8` | ✓ | | | 50.81% |
| `acc8` | | ✓ | | 35.24% |
| `vel8` | | | ✓ | 60.65% |
| `acc8vel8` | | ✓ | ✓ | 78.68% |
| `jerk8acc8` | ✓ | ✓ | | 40.16% |
| `jerk2vel8` | ✓ | | ✓ | 63.93% |
| `jerk4vel8` | ✓ | | ✓ | 79.50% |
| **`jerk8vel8`** | **✓** | | **✓** | **84.42%** |
| `jerk8acc2vel8` | ✓ | ✓ | ✓ | 63.11% |

### 4.1. Feature Metric Approach

When deciding what feature metric would be suitable, we've compared many different models trained for 1 iteration and chose the metric of the model with best test accuracy. From the table above, we've found angular velocity data measured from gyro sensor and jerk data were the significant classifiers in this approach. In conclusion, using `jerk8vel8` model was the best.

### 4.2. Model Complexity Approach

When we construct HMM model with too many states, there can be over-fitting issues that our model only estimates trained data accurately but not test data. By differing number of observation states, we've found out the model having the best complexity would also be `jerk8vel8`.

## 5. Conclusion

Understanding human behaviour using sensor models needs temporally learning models since the dataset is also time-serial. Many other works determining human action recognition uses RNN models like LSTM, probabilistic models like particle filter or in hybrid. We've constructed HMM model simply learning the observation patterns from HAR dataset produced by Jorge L, *et al*.

Dataset includes 1,214 instances of 12 human actions and provides acceleration and angular velocity vectors. Feature extraction in acceleration and angular velocity vector was to splitting its direction into 8 sections, numbered them and use to define observation states. With empirical approaches, we've figured out `jerk8vel8` having appropriate model complexity and high test accuracy as well.

We've trained this model with maximum training iteration of 50 and `jerk8vel8iter50` showed remarkable performance of 92.62% as the test accuracy and 24623.64s as the training time. In future works, we'll improve our model with numerous instances created from more experiments and use other additional sensors to extract various states features.

## References


[1] Adam Geitgey, "How to break a CAPTCHA system in 15 minutes with Machine Learning", Dec. 2017

[2] Adam Geitgey, "The reCAPTCHA Project – Carnegie Mellon University CyLab". *www.cylab.cmu.edu. Archived*, Jan. 2017

[3] yunjey, "Image Captioning", *GitHub pytorch-tutorial*, Nov. 2018

[4] Byung-Jun Yoon, "Hidden Markov Models and their Applications in Biological Sequence Analysis"

[5] M. Gales and S. Young "The Application of Hidden Markov Models in Speech Recognition". *Foundations and Trends in Signal Processing, Vol.1, No.3*, 2007

[6] Jorge L. Reyes-Ortiz, Davide Anguita, Alessandro Ghio, Luca Oneto and Xavier Parra, "Smartphone-Based Recognition of Human Activities and Postural Transitions Data Set" , 2015

[7] Jorge L, *et al*. "Human Activity Recognition on Smartphones with Awareness of Basic Activities and Postural Transitions" *ICANN 2014, LNCS 8681, pp.177–185*, 2014

[8] Rubén San-Segundo, Julián David Echeverry-Correa, Christian Salamea, José Manuel Pardo "Human activity monitoring based on hidden Markov models using a smartphone" *IEEE Instrumentation and Measurement Magazine*, 2016

[9] Daniel Jurafsky and James H. Martin. "Hidden Markov Models" *Speech and Language Processing*, 2019

[10] Wikipedia, Jerk (physics)

[11] API Documentation of sklearn, model selection, train test split