Exploring the Use of Machine Learning Techniques for Finger Vein Biometrics

Nikita Albert
College of Engineering and Computer Science
Syracuse University
Syracuse, NY, USA
nialbert@syr.edu

***Abstract*— As technology continues to become ever more sophisticated, intertwined with day-to-day life and unique to the individual, the need for proper authorization and security grows with it. Although increasingly sophisticated technology allows for increasingly sophisticated security solutions, it also allows for increasingly sophisticated and unique ways for malicious actors to attempt to bypass said security solutions. With advances in recent years, biometric technology and systems are increasingly used to augment security systems when individual identification and authorization is critical. In this paper, we will explore how finger vein images and machine learning techniques may be used together as an effective biometric recognition system.**

I. INTRODUCTION

Advancements in modern computing technology has allowed for the development of increasingly novel, powerful and accessible devices that continue to become ever-more entwined with the facets of ordinary daily life. Owning a smartphone is a near necessity of modern life and is the nexus of every aspect of an individual's life; Smart devices, such as Amazon Alexa or Google Home, are present in more and more homes; Cloud computing is no longer reserved for businesses or organization, and is used by almost everyone to back-up their digital lives.

This increased interconnectedness has been a boon to modern society, however, has raised concerns over privacy, security and access, especially when malicious actors have ease-of-access to greater computing power to bypass existing safeguards. For some situations that necessitate ensuring an individual or certain group of people have access to some system, biometric systems have been implemented as an authorization system or security system augment to great effect. Financial institutions use voice biometrics to help verify whether the person on the phone is who they claim to be; Apple introduced using facial recognition to unlock their smartphones with *Face ID* on the iPhone X; Finger-print and iris biometrics are often used to authorize access to secure sites.

These systems are implemented and often successful because of the inherent principle behind them: certain biological and physical measurements are relatively fixed and unique enough to be used to distinguish between different people and identify specific individuals using past measurements. Apple, for example, claims that, with *Face ID*, the chance of someone unlocking someone else's phone is 1 in 1,000,000 [1]. Despite such great performance with said biometric system, other systems and biometrics may not be as secure and could be prone to spoofing or prone to having too many false positives or negatives. In 2015, for example, the U.S. Office of Personnel Management database was hacked and up to 5.6 million fingerprint images were stolen [2]. This type of breach is dangerous as this provides all the necessary info needed to construct a spoofed fingerprint [3] and relatively easy to use to fool some security systems that rely on fingerprint recognition [4].

Recent research in vascular pattern recognition has yielded impressive results and displays advantages over other biometrics [5]. Researchers found that when sending near-infrared light through a finger, palm or hand, that an image of the vein topology can be captured as the light will be absorbed by hemoglobin where veins are found to create dark spots and the rest of the light will pass through creating the light contrast. As capturing a vein pattern image relies on scanning a part of a live human being and is incredibly difficult to forge, a recognition system based on this biometric can offer greater security and confidence than other contemporary biometric systems.

In this paper we will explore how we may use machine learning techniques on a database of finger-vein images to create a finger-vein recognition system based on image classification.

II. DATASET

The dataset we will be working with is the set of finger-vein images from Shandong University's SDUMLA-HMT database [6]. This dataset is composed of finger-vein images of 106 individuals' index, middle and ring fingers on both hands repeated 6 times, for a total of 3,816 images in *.bmp* format and 320x240 pixels in size.

From a cursory overview, we see that the dataset is of reasonable, workable quality that is already in a standardized format and will allow for smoother preprocessing.

III. EXPERIMENT(S) OVERVIEW

In this paper, the experiment we conduct will actually be a series of exploratory experiments to determine the viability of which methods may be effectively applied to finger-vein images and how to best improve performance. In our first experiment, *Experiment A*, we will construct

machine learning models on a training sample of the dataset to identify which model approach works best.

Researchers in Dongguk University, Korea [7] proposed a method of preprocessing the finger-vein images to reduce the complexity of CNN learning by inputting only a certain region-of-interest of the vein topology for learning and classification. We will take a similar approach by extracting the region-of-interest of our sampled images and using them to train and evaluate models in our second experiment, *Experiment B*.

We may run into the issue where our models are either overfitted or not trained enough, due to the fact that our sampled and region-of-interest datasets are relatively small, containing 636 images each. In our third experiment, we will use the Keras library's *ImageDataGenerator* to augment our preprocessed dataset to build an augmented dataset with many more training samples. In our third experiment, we will train and evaluate models on said augmented dataset.

IV. MODELS

A. Overview

We will construct models of three distinct types in our experiments and evaluate their efficacy on the respective testing sets; an ensemble learning classifier, a probabilistic classifier, and a classifier built on transfer learning with the VGG-16 neural network. Specifically, our ensemble learning classifier will be a *Random Forest* model and our probabilistic classifier will be a *Gaussian Naïve Bayes* classifier.

B. Random Forest Classifier

The *Random Forest* classifier is an ensemble learning model that overcomes the common shortfalls faced by single *decision tree* models by making a model out of multiple *decision trees*. In training, each tree in a *Random Forest* model learns from a random sample of the datapoints, that are drawn from the overall data with replacement. Then at test, the predictions of all individual trees are averaged to make the final prediction. This approach, similar to following the proverbial "wisdom of the crowd," allows for minimizing the risk of a bias, highly variant, and overfitted overall model even if the individual sub-models may potentially be biased, overfitted, or generally sub-optimal.

For our experiments, we will be using the *RandomForestClassifier* algorithm, from the Scikit-learn library, to construct our models.

C. Naïve Bayes Classifier

The Naïve Bayes classifier model, however, follows the Bayes' Theorem to find the probability of an event occurring given the probability of another event that has already occurred. It is a wholistic probabilistic classifier that essentially follows below representation of class probability:

$$P(c_i|x_0,\ldots,x_n) \propto P(x_0,\ldots,x_n|c_i)P(c_i)$$
$$\propto P(c_i)\prod_{j=1}^{n} P(x_j|c_i)$$

Fig. 1. Naïve Bayes Class Probability representation

The model, however, assumes that the values of each attribute are completely independent and do not interact. Although this a strong assumption to make, the model performs well on certain data, where this assumption does not hold. This model is often used in uses cases such as face recognition, medical diagnosis, etc., and we believe its viability for finger-vein recognition is worth exploring.

For our experiments, we will be using the *GaussianNB* algorithm, from the Scikit-learn library, to construct our models.

D. Transfer Learning with VGG-16

In 2012, researchers at the University of Toronto proposed a deep Convolutional Neural Network for image classification that achieved a test error rate of 15.3% in the ILSVRC-2012 competition [8].

Although the model (AlexNet) is relatively simple compared to those used today, it managed to achieve then groundbreaking performance with an error-rate 50% better than the competition's runner-up, which used a Fisher Based Features model [9]. Since then, practitioners have made significant progress in utilizing deep learning methods and models to beat the baseline set by the researches at the University of Toronto and AlexNet.

As our experiments can be viewed as image classification tasks, we will also use and evaluate a deep learning model in them. For a deep learning model to achieve satisfactory general results, however, it needs to evaluate an extraordinary number of parameters. To establish said parameters, the model will need to train on an extraordinary amount of data. AlexNet, for example, was developed and trained on dataset of over 1.3 million images.

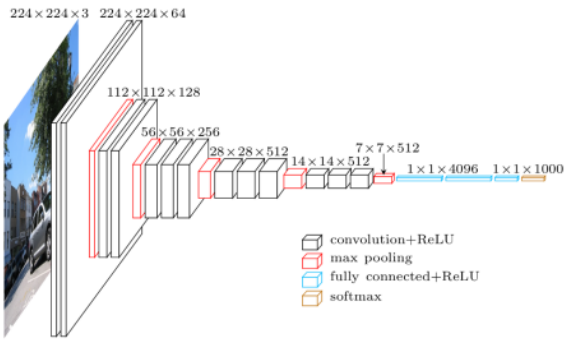

Fig. 2. VGG Net Architecture

To avoid building a poor performing model from scratch, especially when we work with a sampled dataset of 636 images, we adapt an existing model to our needs via transfer learning. The model we adapt is the VGG-16, a very deep convolutional network, proposed by researchers at the University of Oxford, that achieved a top-5 test of error rate of 7.0% against the 2014 ILSVRC dataset [10]. To adapt our model to our data, however, we need to perform some *transfer learning*. To do this, we modify the input layer to accept our finger-vein images, freeze every layer except the last four, and fit the model to our dataset so that the final decided layers are finely tuned to it.

V. PREPROCESSING

A. Restructuring

Currently the SDUMLA-HMT finger-vein image dataset is in the following structure:

- [Finger Vein Directory]
  - 001
    - Left
      - Index_[1…6].bmp
      - Middle_[1…6].bmp
      - Ring_[1…6].bmp
    - Right
      - Index_[1…6].bmp
      - Middle_[1…6].bmp
      - Ring_[1…6].bmp
  - …
  - 106
    - Left
      - Index_[1…6].bmp
      - Middle_[1…6].bmp
      - Ring_[1…6].bmp
    - Right
      - Index_[1…6].bmp
      - Middle_[1…6].bmp
      - Ring_[1…6].bmp

Fig. 3. Original Dataset Structure

As we will only be using the left-hand index finger vein images in our experiments, we will first sample the desired images from the original dataset and store them in a new directory structured towards our needs. We will also use this restructuring as an opportunity to generate our training and testing sets. The first four left-hand index finger vein images will be used as our training images and the last two will be used as our test images. We visualize our new directory structure below in *Figure 4.*

- [Finger Vein Directory - Restructured]
  - Train
    - 001
      - Index_[1…4].bmp
    - 002
      - Index_[1…4].bmp
    - …
    - 106
      - Index_[1…4].bmp
  - Test
    - 001
      - Index_[5…6].bmp
    - 002
      - Index_[5…6].bmp
    - …
    - 106
      - Index_[5…6].bmp

Fig. 4. Sampled Images Dataset Structure

B. Region-of-Interest Extraction

Now that we have our training and test sets of base images established, we will generate our training and test sets of region-of-interest images by processing our sampled images.

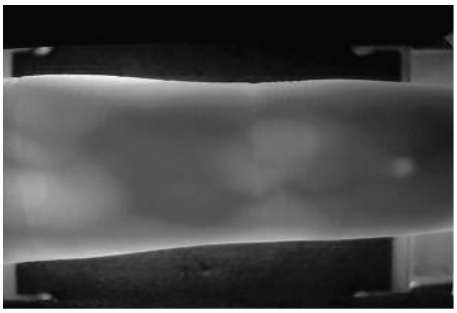

Fig. 5. Visualization of typical sampled image

Each of our sampled images are similar to *Figure 5*, above: a 320x240 black and white image of a finger and its vein topology, captured by scanning the finger by near-infrared light.

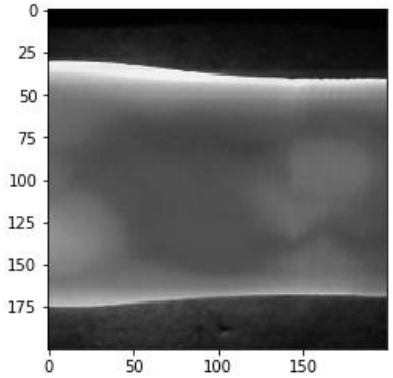

Fig. 6. Visualization of Image Preprocessing Step: Cropping

First, we crop our image to exclude the corner guiding markers and ends of the fingers, where we expect poor image quality due to light leaking in from outside of the scanner. We crop each image from x-position 40 to x-position 240 and y-position 20 to y-position 220, leading to a resultant draft image, exemplified in *Figure 6*.

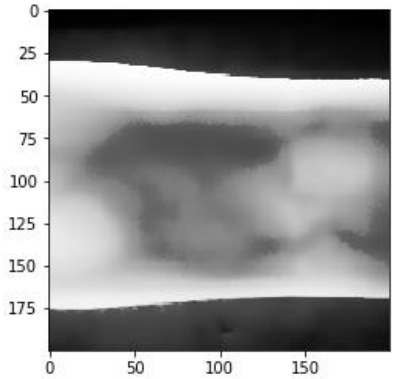

Fig. 7. Visualization of Image Preprocessing Step: Image Enhancement

We then enhance our cropped draft image by improving its contrast and filtering it to reduce noise by applying

OpenCV's *equalizeHist* and *bilateralFilter* functions on it, respectively.

When we attempted to perform edge-detection on our cropped and enhanced images to get the bounds of the fingers, we ran into inconsistent results when applied methods on the entirety of the image. To overcome this, we split the image in half horizontally and apply edge-detection methods to find the upper and lower bounds respectively.

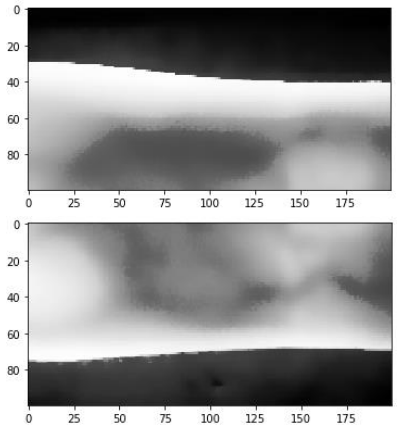

Fig. 8. Visualization of Image Preprocessing Step: Splitting

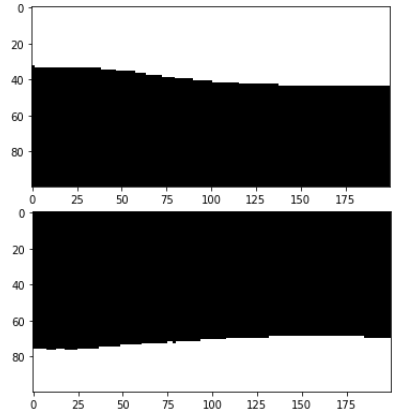

Fig. 9. Visualization of Image Preprocessing Step: Preliminary Masking

We achieved straightforward and good results using OpenCV's *Canny* edge detection. To ensure we have solid masking, we flood fill outside the boundary with black and inside the boundary with white before we concatenate both halves together again.

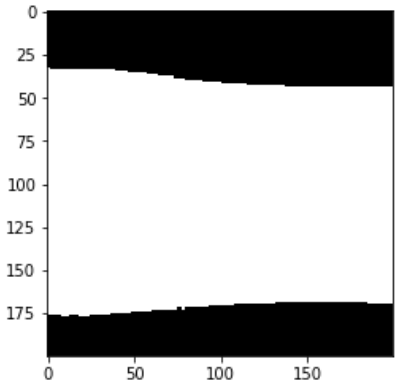

Fig. 10. Visualization of Image Preprocessing Step: Final Masking

We inverse the colors of our concatenated mask, so that the region we desire to keep is white and the region we desire to exclude is black. We then apply the mask onto our enhanced image to keep finger-vein ROI, take the bounding box coordinates of the ROI, and use OpenCV's *warpPerspective* function to expand the ROI to fill the entirety of the 200x200 pixel space.

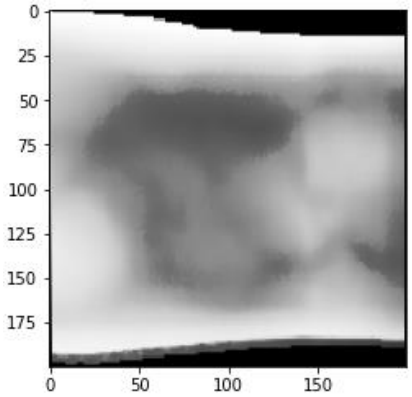

Fig. 11. Visualization of Image Preprocessing Step: Final ROI image

C. Image Augmentation

To augment our region-of-interest finger-vein images, we use the *ImageDataGenerator* class from the Keras library. With a few lines of code, we are able to artificially create multiple, unique, processed training images from one. To build our augmented training set, we will read in each region-of-interest image and augment them into ten new images that will be some combination of being vertically shifted, horizontally shifted, rotated, zoomed in/out or of varying brightness. We will also generate an augmented testing set to evaluate how our models, trained on the augmented images, fare. For the testing set, however, we will only make three augmented images for each region-of-interest one.

To ensure experimental fidelity and validate that augmentation helps our models make more general evaluations when making predictions, we will evaluate how our models, in Experiment C, perform on both the region-of-interest and augmented testing sets.

VI. RESULTS

A. Experiment A

Base Image Models				
Model	Accuracy	Precision	Recall	F1
Random Forest	77.358%	73.854%	77.358%	73.658%
Naïve Bayes	1.887%	0.638%	1.887%	0.773%
VGG-16	90.566%	87.170%	90.566%	88.023%

Table 1. Experiment A metrics

In our first experiment, we see some interesting results. Our adapted *VGG-16* model achieves the best performance,

with an accuracy of 90.566%. *The Random Forest* model achieves the next-best performance, with an accuracy of 77.358%. The *Naïve Bayes* model, however, only manages to achieve an accuracy of 1.887%.

The significantly poor performance of the *Naïve Bayes* model is not surprising. The model makes predictions based on probabilistic patterns that it finds during training; As it only learned on four samples per class, for 106 classes, the model was significantly undertrained.

B. Experiment B

Processed Image Models: Full Sample				
Model	Accuracy	Precision	Recall	F1
Random Forest	64.151%	61.952%	64.151%	60.657%
Naïve Bayes	5.660%	3.588%	5.660%	3.285%
VGG-16	78.774%	76.855%	78.774%	75.842%

Table 2. Experiment B metrics

Surprisingly, our *Random Forest* and *VGG-16* models, trained and evaluated on our preprocessed region-of-interest images, perform worse than their counterparts in *Experiment A*. Our *Random Forest* model achieves a 64.151% accuracy and our *VGG-16* model achieves a 78.774% accuracy; Degradations of 17.07% and 13.02%, respectively. Our *Naïve Bayes* model achieves almost a 200% increase in performance; However, this increase is not impressive as the model's accuracy is only 5.660%.

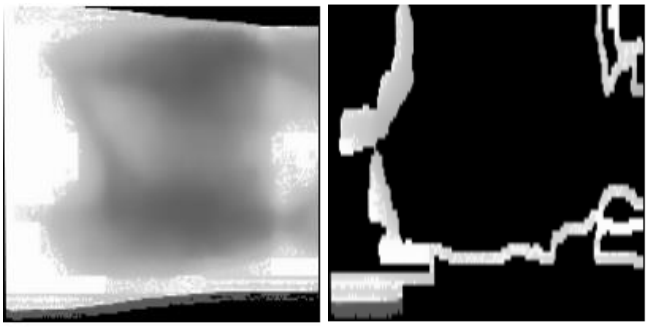

Fig. 12. Extracted ROIs of Individual 036's Finger-Vein Images. Good vs. Bad (Left, Right)

We took a look at the training and testing images of our mispredictions to understand what could cause a drop in performance and we observed two potential causes.

The most obvious cause is that our region-of-interest extraction algorithm severely underperforms and yields us poor resultant images. In *Figure 12,* we see how our algorithm yields us the desired ROI for one of Individual 036's finger-vein images but fails to for the next image. As the adage goes: garbage in, garbage out.

The second cause may be due to the fact that the base finger-vein images are of poor quality. For example, we see how washed out an image can be and how the vein topology may be difficult to make out when we visualize one of the finger-vein images for Individual 061, in *Figure 13.*

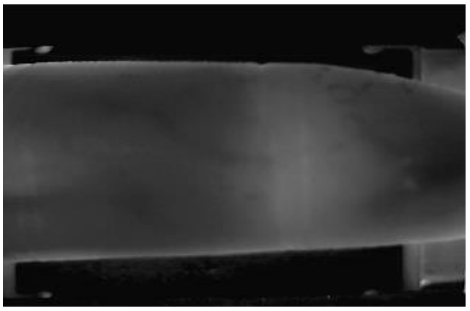

Fig. 13. Base Finger-Vein Image: Individual 061

Fig. 14. ROI Finger-Vein Image: Individual 061

For Individual 061, we manage to extract and enhance their finger-vein region-of-interest, as seen in *Figure 14;* However, such enhancement may not be enough, as we still have a washed-out segment on the right side of the image. This suspicion is confirmed by how the researchers at Dongguk University referred to this dataset as the "low-quality" dataset due to the increased shading and quality inconsistencies not found in other datasets they worked with [7]. So, when our models were trained on the overall base images, they could have learned from information that is no longer available, such as the overall shape of the finger, positioning, etc, and rely on poor quality vein topology.

Processed Image Models: Quarter Sample				
Model	Accuracy	Precision	Recall	F1
Random Forest	80.0%	77.667%	80.0%	76.8%
Naïve Bayes	46.0%	43.480%	46.0%	39.725%
VGG-16	90.0%	86.667%	90.0%	87.20%

Table 3. Experiment B metrics for models constructed on limited samples

To validate our hypothesis that better quality images would lead to better model performances, we choose to retrain and retest our models on a smaller sample of images. We found that our processed images for the first 25 individuals were of relatively good quality, without any washing out or poor region-of-interest extraction and move forward with them. As we see in *Table 3,* we see significant gains in performance, especially with our *VGG-16* model that achieves 90% accuracy.

C. Experiment C

Augmented Image Trained Models		
Model	ROI Test Set Accuracy	Augmented ROI Test Set Accuracy
Random Forest	70.283%	71.226 %
Naïve Bayes	61.321%	57.075%
VGG-16	88.207%	88.679%

Table 4. Experiment C metrics

In our third experiment, we see how using data augmentation to make multiple augmented images out of each preprocessed image, pays dividends in model training and subsequent model performance. When we applied our models, trained on augmented ROI images, on the test set containing just the extracted ROI test images: Our *Random Forest* model achieved predictions with an accuracy of 70.283%; Our *Naïve Bayes* model achieved predictions with an accuracy of 61.321%; Our *VGG-16* model achieved predictions with an accuracy of 88.207%. Overall, model improvements of 8.72%, 982%, and 11.97%, respectively.

When we applied our models on the test set contained the augmented ROI test images, our *Random Forest* and *VGG-16* models achieved slightly more accurate predictions, whereas our *Naïve Bayes* model dipped in performance.

VII. CONCLUSION

From our experiments, we can conclude that machine learning algorithms and methods may be applied on finger-vein images to use them as a viable biometric. Although different algorithms and models yield varying results, we can state confidently that deep learning models offer the most promising performance. We saw, for example, how our transfer-learned VGG-16 model outperformed our other two models, with relatively good performance, even when the quality of our dataset degraded in *Experiment B.*

We can also conclude that data augmentation can be a powerful tool to make small datasets viable for training machine learning models. We saw how each of our models in *Experiment C* saw gains in performance, compared to their counterparts in *Experiment B*, because they were trained on multiple variations of the images used in *Experiment B*, even though some of the images were of poor quality.

As such, we believe that with a quality apparatus to capture good quality finger-vein images, some data augmentation, and some machine learning know-how, a powerful, accurate and secure biometric system may be developed.

REFERENCES

[1] Apple, *About Face ID advanced technology*. Accessed on: Feb 8, 2020. [Online] Available: https://support.apple.com/en-us/HT208108

[2] A. Peterson, *OPM says 5.6 million fingerprints stolen in cyberattack, five times as many as previously thought*, The Washington Post, Sep. 8, 2015. Accessed on: Feb 13, 2020. [Online]. Available: https://www.washingtonpost.com/news/the-switch/wp/2015/09/23/opm-now-says-more-than-five-million-fingerprints-compromised-in-breaches/

[3] Synaptics, *Protecting Against Fingerprint Spoofing in Mobile Devices*, Synaptics, 2016. Accessed on Feb 15, 2020. [Online]. Available: https://www.synaptics.com/sites/default/files/sentrypoint-anti-spoofing-wp.pdf

[4] A. Preuschat, *Watch Out, Your Fingerprint Can Be Spoofed, Too*, Wall Street Journal, Feb. 24, 2016. Accessed on Feb 15, 2020. [Online]. Available: https://blogs.wsj.com/digits/2016/02/24/watch-out-your-fingerprint-can-be-spoofed-too/?mod=ST1

[5] FBI, *Vascular Pattern Recognition*, FBI. Accessed on Feb. 5, 2020. [Online]. Available: https://www.fbi.gov/file-repository/about-us-cjis-fingerprints_biometrics-biometric-center-of-excellences-vascular-pattern-recognition.pdf/view

[6] *"SDUMLA-HMT Database,"* Shandong University, 2010. Accessed on Feb. 3, 2020. [Online]. Available: http://mla.sdu.edu.cn/info/1006/1195.htm

[7] Hong, Hyung & Lee, Min & Park, Kang. (2017). Convolutional Neural Network-Based Finger-Vein Recognition Using NIR Image Sensors. Sensors (Switzerland). 17. 10.3390/s17061297.

[8] Krizhevsky, Alex & Sutskever, Ilya & Hinton, Geoffrey. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems. 25. 10.1145/3065386.

[9] N. Gunji, T. Higuchi, K. Yasumoto, H. Muraoka, Y. Ushiku, T. Harada, and Y. Kuniyoshi, *Scalable Multiclass Object Categorization with Fisher Based Features*, Intelligent Systems and Informatics Lab, University of Tokyo, 2012. Accessed on: Feb 18, 2020. [Online]. Available: http://image-net.org/challenges/LSVRC/2012/isi.pdf

[10] Simonyan, Karen & Zisserman, Andrew. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv 1409.1556.