UKRAINIAN CATHOLIC UNIVERSITY

BACHELOR THESIS

Accuracy And Bias Of Selfie Detection On Open Data

Author:
Natalia-Yana SHPOT

Supervisor:
PhD Miriam REDI

A thesis submitted in fulfillment of the requirements
for the degree of Bachelor of Science

in the

Department of Computer Sciences
Faculty of Applied Sciences

УКРАЇНСЬКИЙ КАТОЛИЦЬКИЙ УНІВЕРСИТЕТ

APPLIED
SCIENCES
FACULTY

Lviv 2020

Declaration of Authorship

I, Natalia-Yana SHPOT, declare that this thesis titled, “Accuracy And Bias Of Selfie Detection On Open Data” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"Done is better than perfect."

Sheryl Sandberg

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Bachelor of Science

Accuracy And Bias Of Selfie Detection On Open Data

by Natalia-Yana SHPOT

Abstract

There are many challenges related to the openness of the Wikimedia Commons image upload platform, and one of them is about making sure to get high-quality content in. Goes without saying, selfies are not precisely the ideal wanted content for a platform whose aim is to represent the world's knowledge through pictorial representations. One way to automatically check the data quality in the domain of computer vision is to design a selfie detector that, given an image, can automatically predict whether it is a selfie or not. Thus in this thesis, we are using state-of-the-art models to create a classifier that, given an image, can say whether the image is a selfie, a person, or neither of that. With such a classifier, it would be easier to automatically detect and scale selfies for Wikimedia or other platforms that have humans in the loop to check the quality of user-generated content. In addition to this we examine whether approaches of our choice show bias in demographics such as race, gender, and age. Furthermore, we will introduce two datasets for our project: one containing selfies, pictures with persons and random pictures, and another containing a smaller set of pictures of persons along with the demographic metadata.

Acknowledgements

I want to show my gratitude to Miriam Redi, who was supporting and actively leading me through my entire work. Also I would like to thank everyone who is always near and believes in me more than I beleive in myself.

Contents

List of Figures

List of Tables

List of Abbreviations

CNN	**C**onvolutional **N**eural **N**etwork
SGD	**S**tochastic **G**radient **D**escent
RMSProp	**R**oot **M**ean **S**quare **P**ropagation
Adam	**A**daptive Moment Estimation
AdaGrad	**A**daptive **G**radients
Adadelta	**A**daptive Learning Rate

0.1 Introduction

0.1.1 Motivation

A tremendous amount of data is appearing every second and is constantly changing the world. With each year, technology, and the ability to collect and make use of data, continues to advance rapidly. However, not all data is equally useful. It is essential for information, brought by data, to be available in sufficient quality so it could be reused.

The Wikimedia Foundation, Inc.[1] is one of those non-profit charitable organizations, which works on bringing free and qualitative educational content to the world. One of their projects, Wikimedia Commons[2], is a media file repository that makes available public domain and freely-licensed images, sound, and video clips to everyone. Also, everyone can upload his content by Wikimedia's image upload platform, thus checking the quality of it is one of Wikimedia's responsibilities and challenges. It is obvious that for this task, they need different robust algorithms to automate processes because of the large amount of freshly uploaded data. Nevertheless, for now, in most cases, such data is reviewed manually.

Nowadays, it became so easy for everyone to take a selfie anytime and anywhere. As selfies gained popularity, this kind of pictures became an issue for the Wikimedia platform as well. For that reason, one of the existing challenges is to be able to recognize selfies from uploaded pictures, taking into account the diversity of data.

0.1.2 Goals

For this thesis, we decided to approach this problem by developing and evaluating a 3-class image classifier, which distinguishes selfies, pictures containing people, and random photos. As the data on Wikimedia is significantly diverse (as it comes from all over the world), the chosen model needs to show adequate performance on any type of it. Thus our second task is to analyze how biased the selected model is on different genders, ages, and nationalities and to determine what categories are the most problematic.

0.1.3 Generalized Overview

Although the idea for this work was based on the Wikimedia case, this topic should be viewed from a broader perspective as it is more general. Selfie detection may also be considered as one of the ways to organize pictures automatically in any data storage, for example, in photo libraries of our devices, where photos are mixed and not distinguished as the ones made by a frontal camera (Selfie folder in iPhone library).

Demographic bias is a separate problem on its own. Fairness in machine learning is a greatly popular and topical subject. Usually, datasets, containing people, are not balanced by demographic categories. Models trained on such data tend to to show inaccurate results when applied to new data containing underrepresented categories. But one can't say that the model is good if it performs well only on typical for the train data appearances, for example if picture is of a "white man". Thanks to the huge variety of pictures on Wikimedia Commons we can have a fairly qualitative anlysis of how biased a model can be on open data.

[1] https://wikimediafoundation.org

[2] https://commons.wikimedia.org/wiki/Main_Page

0.1.4 Structure

Chapter 2: Related Work

In this chapter we will give a general overview of the previously conducted researches that are relevant to selfies and bias.

Chapter 3: Background Information

This chapter introduces the models, evaluation metrics and experimental pipeline details used for this work

Chapter 4: Datasets

This chapter describes what datasets we collected, how and for what purposes we did that.

Chapter 5: Evaluating Classifier

In this chapter information about experimental setup and experimental results along with their interpretations is provided.

Chapter 6: Bias Analysis

This chapter is about the ways we were analysing selected models for demographical bias and overview of the results we obtained.

Chapter 7: Conclusion

This is the final chapter which summarizes the results of this thesis and proposes future imrpovements that will be partially implemened before the thesis defense.

0.2 Background Information

0.2.1 The Models

Inception

The Inception network was an important milestone in the development of CNN classifiers. Before this model was introduced, the most popular CNNs used to improve performance simply by stacking convolution layers deeper and deeper. On the other hand, the Inception network was heavily engineered, and instead of going deeper, it goes wider.

For the first time, the inception module has been described in the paper "Going Deeper with Convolutions" by Szegedy et al. in 2015[14]. This innovative module is a block of parallel convolutional layers with filters of 3 different sizes and a 3×3 max-pooling layer, the results of which are extracted and concatenated before it is fed to the next layer. Three filters are needed to capture features of multiple scales.

For this work, we decided to use the InceptionV3 model. With 42 layers, the lower error rate is obtained and made it become the 1st Runner Up for image classification in ILSVRC[3] in 2015. It is not the most recent Inception model, yet it remains popular and widely used in research and production.

There are many typos in the Inceptionv3 paper[15], which lead to confusion between Inception versions. Nevertheless, in Inceptionv4 paper[13], we received a clear description of the versions' differences: "The Inception deep convolutional architecture was introduced as GoogLeNet in (Szegedy et al. 2015a), here named Inception-v1. Later the Inception architecture was refined in various ways, first by the introduction of batch normalization (Ioffe and Szegedy 2015) (Inception-v2). Later by additional factorization ideas in the third iteration (Szegedy et al. 2015b) which will be referred to as Inception-v3 in this report."

EfficientNet

EfficientNet was introduced in 2019 by Tan et al. in the paper "Efficientnet: Rethinking model scaling for convolutional neural networks" [16], which not only focuses on improving the accuracy, but also the efficiency of models. Although earlier researches have also been done in the direction of reducing the number of parameters and FLOPS so that models can be run on mobiles, it is the first time where huge gains in parameter reduction and FLOPS cost along with significant gain in accuracy is shown.

Compared to other models with similar accuracy on the ImageNet, EfficientNet is much smaller. For example, the ResNet50 has over 23 million parameters, but it still underperforms the smallest EfficientNetB0, with slightly more than 5 million parameters in total.

The Efficient network's baseline consists of Inverted Residual Blocks (used in MobileNetV2) with a Squeeze and Excitation optimization.

There are three scaling dimensions of a CNN: depth, width, and resolution. The observation made in the paper was about the critical importance of balancing all

[3] http://www.image-net.org/challenges/LSVRC/

three dimensions of a network during CNN's scaling for getting improved accuracy and efficiency. For this purpose, Compound Scaling was proposed, where a user-specified compound coefficient ϕ is used to scale network width, depth, and resolution uniformly.

This scaling technique was used to produce different versions of EfficientNet. Starting from the smallest EfficientNetB0 configuration to the largest EfficientNetB7, the model's accuracy is constantly increasing while maintaining a relatively small size.

The authors showed that when the EfficientNet backbone is used, the performance improves not only for but for other computer vision tasks as well.

0.2.2 The Classifier

Evaluation Metrics

The most common metric to evaluate the performance of a classification model is *classification accuracy*. Accuracy is the proportion of examples that were predicted correctly, divided by total amount of predictions that were made.

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}} \tag{1}$$

The result of accuracy should be compared to the result achieved by a model that makes random predictions. It is a minimal accuracy expected from each untrained model based on the number of classes and their balance. As there are three equally significant classes, it leads to the conclusion that the baseline accuracy for this classification task is 30%.

Accuracy is a good place to start but still is insufficient for proper evaluation of the classification model. We can check overall model performance with the help of accuracy but cannot make any conclusions about the nature of such a result. A *confusion matrix* is more informative as it shows ways in which the chosen classification model is confused when it makes predictions. With the confusion matrix, we can examine which classes are being predicted correctly, which incorrectly, and what type of errors are being made. In a confusion matrix, each row represents the actual class of the data in that row, and each column represents the predicted class. A value in the cell (i, j) is a count or a percentage of predictions made for ith class that were predicted as jth class. The cells on the diagonal represent correct predictions, where a predicted class aligns the expected one.

With the help of a confusion matrix, we can evaluate *Type I* and *Type II errors*. In statistical hypothesis testing, a Type I error is the rejection of a null hypothesis that is actually true (also known as a "false positive"). In contrast, a Type II error is the acceptance of a false null hypothesis ("false negative"). Although we cannot completely eliminate both errors, we need to decide what type of error is worse for a particular task and work on minimizing it. The primary purpose of the selfie classification task from the view of the Wikimedia problem is to detect and get read of all selfies, which are one example of bad data. As all data, which was chosen for deletion is being additionally reviewed, it is more important not to miss a selfie than to misclassify non-selfie as the one. Thus false negative predictions of selfies are less acceptable, and we have to focus on reducing Type II error for this class.

Since we are interested in minimizing False Positive Rate, we should take into account sensitivity. This metric answers the question how "sensitive" is the classifier in the detection of true positive instances. It is also known as True Positive Rate or Recall.

$$\text{Sensitivity} = \frac{\text{True Positive}}{\text{True Positive + False Negative}} \tag{2}$$

In the case of selfie detection, it gives us the percentage of correctly predicted selfies of all actual selfies. We will receive sensitivity for each class on the diagonal of the confusion matrix if we normalize it by rows. We aim to maximize sensitivity for selfies.

Last but not least is the visual evaluation of predictions. By exploring predicted pictures, we can find out what types of pictures are prone to be misclassified and gain some valuable insights we could notice only by seeing the real pictures. It is possible to represent the pictures in many ways. The simplest way is to show a random subset of the predicted pictures. But for complete understanding, we can sort them by the value of the prediction made. In this way, it becomes possible to display pictures the model was the most certain about. Or vice versa, pictures sorted in the opposite direction may be displayed so that we will find out with what types of pictures model encounters troubles. Finally, we can go deeper and explore all these kinds of predicted pictures representation for each class separately for easier comprehension.

Description of The Experiments

These models were already trained on many datasets before, one of which is Imagenet[5]. Imagenet is an extremely large dataset as it has over 14 million pictures divided between 21,841 subcategories. Even though models are usually trained only on 1000 categories, it takes hundreds of hours on multiple high powered GPUs to properly train a model on such a dataset. The first step was to determine how the chosen models perform on our dataset with the help of transfer learning. Transfer learning means that we make use of this already pre-trained model for the new data. We transfer the weights responsible for feature extraction and replace the top layers of the model with the ones which correspond to the classification we want to perform. It is a popular methodology as the effectiveness of transfer learning is supported by a vast amount of evidence. It was shown that "transferring features even from distant tasks can be better than using random features" and that "initializing a network with transferred features from almost any number of layers can produce a boost to generalization that lingers even after fine-tuning to the target dataset"[17].

There are several steps required to adjust the model for our task. First of all, we need to freeze the transferred part of the model and train only the last layers we added. To do so, we are seeking for the right combination of hyperparameters for the most optimal result. After finding hyperparameters that fitted the most, we train the last layers of the model to some point. The number of epochs may also be considered as a separate hyperparameter. Even at this stage, it is essential to avoid early overfitting as a model may get stuck in local minima. The next step is to perform fine-tuning. The initial layers of a model encode generic, reusable features, like curves and edges. The further the layers are situated, the more specific features they encode. The number of layers we unfreeze for fine-tuning is yet another decision to make. We didn’t tune this parameter and chose to train the top 2 inception blocks, as was suggested in the official example. Therefore we froze the first 249 layers and unfroze the rest.

After exploring a model with transfer learning, we remove pre-trained weights, unfreeze all layers, and with freshly tuned hyperparameters retrain it from scratch. By performing that, we intend to find out if our dataset alone carries enough information to cover the needs of our particular task. That is, how well the model can learn the required features without weights pre-trained on the ImageNet. The comparison of both model types will be described later.

0.3 Related Work

0.3.1 Selfies

Selfies are often used in scientific works that explore them alone. We can find works about quality of selfies, for example "How to take a good selfie"[7], about selfie-related behaviour [3], personality-related cues in selfies [11], or about characterization of selfie contexts [4], etc. Nevertheless selfie is a highly common type of photography, not many works can be found which make use of this specific pictures for other purposes.

One of the most notable domains where selfies are considered is a security domain. Authentification with a selfie is an important and convenient mean of verifying identity for secured access on mobile devices. There is a book about selfie biometrics [12] where a clear overview and presentation of recent advances and challenges in this field are provided along with numerous selfie authentication techniques. Face recognition security systems face vulnerabilities such as printed photos, replayed videos and 3d mask attacks. Therefore, the identification of selfies' genuineness is an important task.

Here is one relevant paper[10] which proposes an anti-spoofing algorithm to detect fake faces from selfies. They use Naïve Bayes classifier algorithm to classify data as fake or real. Our work is similar in some ways, because it is also about classification with selfies. But the classification alone is the opposite as we concentrate on detecting selfies in our work, and authors of this paper concentrate on detecting pictures that are not selfies.

One more of the notable researches in this area was conducted by Annadani et al.in 2016[1], where they leveraged the idea of body position when person makes a selfie. Specifically, they were detecting head, shoulders and hand angels to infer whether the photo is a selfie. While it seems to be a good approach with noticeable results we want to develop the model which will also cover a broader scope of selfie settings. We suppose that we can achieve results that are decent enough with the help of deep neural networks alone.

0.3.2 Bias

In the past five years fairness in machine learning is gaining more and more attention. It is all because the usage of machine learning techniques became more prevalent in peoples' everyday lives. For that reason it is crucial to watch for and eliminate bias in the models that are developed.

There are lots of papers concerning bias and fairness that are published every year. But nevertheless it seems that this problem will not be completely solved in the near future. For example, only a few weeks ago the Time magazine published an article about that[4]. Most of the existing datasets are biased because of multiple reasons. For example, some data about company hiring might contain a lot of gender bias because historically a lot of companies were treating women candidates unfairly. Consequently, any data science model trained on that data, will have a lot of bias irrespectively to how good it is.

That is because the source of the problem is dataset itself. It is the reason why scientific community are now concentrated on providing quality datasets with balanced

[4] https://time.com/5520558/artificial-intelligence-racial-gender-bias/

representation of every protected group. That is, a group of people based on protected property such as gender, race, age, faith and others.

One such example relevant to our domain was introduced by Karkkainen et al. recently in 2019[8]. They collected over a 100,000 images of people faces from Flickr[5]. The main feature of this dataset is that it is race-balanced and thus it does not have bias. Additionally, they also provided a metadata for each image regarding gender and age, which are also protected groups. Unfortunately, we cannot use that dataset because our problem is focused on classifying selfies rather than just cropped people faces.

[5] https://www.flickr.com/

0.4 Datasets

To the best of our knowledge, no publicly available dataset was suitable for training the classifier with selfies, person, and random images. So as a part of our contribution, we collected and made publicly available the "Selfie Classification Wiki" dataset with those three classes.

`https://www.kaggle.com/yasiashpot/selfie-classification-wiki`

Another major part of our work is to check the demographic bias of selected models. To do it, we needed a dataset with peoples containing demographic information such as race, skin color, age, sex, et cetera. Fortunately, our selfie part of the "Selfie Classification Wiki" dataset contained such useful metadata. Moreover, on the Wikimedia, there is a smaller dataset of people annotated with such data. While its size is not sufficient for our model training, it will serve its purpose for testing the model bias afterwards. So we merged those two parts into the "People Demographics Wiki" publicly available dataset.

`https://www.kaggle.com/yasiashpot/portraits-with-demography-dataset`

0.4.1 Selfie Classification Dataset

"Selfie Classification Wiki" dataset consists of pictures categorized by three classes - selfie, person, and random. Pictures of the "person" class have at least one real person on them, and "random" pictures are anything else but selfies or pictures with people.

General Characteristics

The dataset itself has 140,400 pictures in total, with 46,800 pictures per each of three classes. Selfies are 306x306 px RGB pictures. The other two classes contain pictures that are of 600 px width and of variable height. As it can be seen from the table below 1, the *selfie_classification.csv* document has filename, class, and test columns, where "test" is to identify pictures that belong to the "person" and "random" classes, which were manually checked. These checked pictures may be used as a test and validation data for more accurate evaluations.

Column	Description
filename	image relative filepath from the root of dataset
class	integer in range from 0 to 2 identifying class of the image. You can see what each label means in Table 2
test	True, if image was manually confirmed that to be a part of specified class

TABLE 1: Content of Selfie Classification

Label	Description
0	selfie class
1	person class
2	random class

TABLE 2: Content of Selfie Classification

Data Collection and Examination

We created the "Selfie Classification Wiki" dataset by combining the following three sources of data:

- randomly selected pictures from Wikimedia Commons
- pictures of categories which had "people" in their name from Wikimedia Commons
- selfie dataset [7]

To the best of our knowledge, we used the only publicly available dataset for selfies. It was also attractive for us because it is annotated by attributes we could use for the bias analysis. There is some amount of standard-size pictures, which were already padded, but a significant amount of all pictures appeared to be cropped so that the face takes up almost all the space of a picture. Although the Selfie dataset is perfect for the purpose it was created, it might not be the best option for the Wikimedia use case. We suppose that not preprocessed and more diverse types of selfies would be a better fit for this purpose, but at this stage, scraping selfies from the internet and manually cleaning them is not justified due to lack of resources.

Pictures for the other two classes were retrieved from Wikimedia Commons' dataset with the use of either a simple "wget" query or CommonsDownloader tool[6] by their names or by categories. For the person class, we downloaded all pictures from categories that had the word "people" in their name.

Since categories on Wikimedia are crowdsourced and does not go through a centralized review process, it was expected for those pictures to be not completely clean. It means that not every picture from the person class has a person on it. Due to the fact that the third "random" category was also automatically collected, there is no way to guarantee that random photos will not contain any picture of a person, even after excluding all human-related categories. As expected, there occurred to be many people on random pictures as well.

Knowing this information, we took and analyzed a representative sample of randomly selected pictures per each class to make sure that the percentage of misclassified images is reasonably small. The size of this sample was computed using the formula.

$$\text{Sample size} = \frac{\frac{z^2 \times p(1-p)}{e^2}}{1 + (\frac{z^2 \times p(1-p)}{e^2 N})} \tag{3}$$

where N is the population size (number of images of a class), e - margin of error, z - Z-score, p - standard of deviation.

[6] https://pypi.org/project/CommonsDownloader/

Having performed a check of the calculated statistically significant amount of random pictures, we came to the conclusion that each class has less than 10% of inappropriate labeled data with 95% confidence and a 5% margin of error rate. Thus, we assumed that this data is good enough to be used for our classification task.

Data Preprocessing

Since we were collecting data from different sources and merging them together, we also needed to make some additional steps to make further preprocessing to allow efficient usage of the dataset. In general, we made the following enhancements:

- renamed files containing POSIX-incompatible characters
- removed gif files
- enforcing balanced dataset
- unifying size of Wikimedia images

So now, let us discuss each of those items in detail.

First of all, we renamed all files containing POSIX-incompatible characters. That was a requirement before we can make our dataset public via Kaggle, which throws an error on every file violating this rule.

Secondly, OpenCV[2] does not yet support processing gif file due. Thus we were required to adjust collection scripts to omit those files.

Furthermore, since every data source had a different number of images, we randomly selected the number of pictures that equals the size of the smallest subset from every source to make sure that the dataset is balanced.

Lastly, we also made sure to download all Wikimedia images with the same width of 600px, preserving the original height. That allowed us to avoid artifacts during resizing because the original aspect ratio was preserved. At the same time, we got an easy-to-use set of unified images.

0.4.2 Dataset With Demographics

Another part of our dataset work was dedicated to creating a dataset to check the bias.

Selfie

As part of our demographics dataset, we included a selfie data [7], which also has some useful manually marked data about person gender, age, and race. Specifically, the metadata can be accessed from *selfie_dataset.txt* file, which has the following relevant fields, described in Table 3.

Please note that those columns are binary flags, so when all columns from the same category are not set, it represents N+1 value for that category. For example, when "white", "black", and "asian" flags from race category are all equal to -1, then the race of this example we treat as "latino/other" race.

Categories	Columns
gender	is_female
race	white, black, asian
age	baby, child, teenager, youth, middle_age, senior

TABLE 3: Selfie Demographics Data

Persons

Another part of the dataset are photos of persons from the Wikimedia Commons. Fortunately, it also offers some useful demographics metadata on around 25,000 pictures of persons. As we mentioned, although it was not sufficient to use these pictures as our training data, they create a large and diverse test dataset for bias check. In particular, all that metadata can be found in *bias_filename.tsv* file, with the information described in Table 4.

Column	Description
itemLabel	person's name
genderLabel	person's sex
ethnicityLabel	person's ethnicity
dobLabel	person's date of birth

TABLE 4: Persons Demographics Data

0.5 Evaluating Classifier

0.5.1 Experimental Setup

We have chosen InceptionNetV3 and EfficientB3 for our experimental work. The first one is an example of something more typically used but still effective, while another is an example of something relevantly new.

Description of Hyperparameters

One of the main challenges in training every model is to select the right hyperparameters. Here is the extensive list of the hyperparameters we tried to tune for the models we worked with:

- Learning rate: 0.0001 - 0.1
- Optimisation: sgd, rmsprop, adam, adagrad, adadleta
- Dropout: 0.2 - 0.6
- Adding learning rate decay
- Adding momentum
- Batch size: 32, 64, 128
- Adding additional Dense layer and number of its units: 514, 1024
- Normalising dataset [-1,1], [0,1]

"The learning rate is perhaps the most important hyperparameter. If you have time to tune only one hyperparameter, tune the learning rate." - Page 429, Deep Learning by Ian Goodfellow, 2016[6].

Learning Rate is tightly connected with the choice optimizer. Rmsprop, Adam, Adagrad, and Adadleta are the optimizers that adapt learning rates for each weight in the network by individual methodologies. The SGD optimizer is a standard stochastic gradient descent we all know about.

Adding and regulating the Dropout layer is one of the first things to do to avoid overfitting. Setting the decay may also be useful for this purpose.

The freshly published research "Rethinking the Hyperparameters for Fine-tuning" by Li et.al.[9] shows that the value of momentum also affects fine-tuning performance, and must not be treated only as a way to speed up the convergence of a model.

Changing batch size does not seem to make a significant impact on performance as long as its value is not too big. For this reason, opted out of tuning this hyperparameter and used 64 for InceptionV3 models and 32 for EfficientNet models, as the usage of a batch size of 64 pictures for EfficientNet tried to allocate more memory than was available.

At first, we were normalizing pixels of each picture from minus one to one, as was being done for the paper with the ImageNet for the InceptionV3. But it led to worse results; therefore, we switched to a standard way to normalize pixels from zero to one.

Data Usage And Preparation

For both InceptionV3 and EfficientNet models, we worked with data in the same way.

It should be noted that we were not working with the whole dataset we created at once, as all the processes would have taken an inefficient amount of time with a single GPU available to us on Kaggle. That is why, depending on a task, we used from 5000 to 45000 of pictures. We usually tuned hyperparameters on 5000 - 15000 pictures and then used 30000-450000 for training the model. Even though the initial dataset is balanced, there is still a possibility that randomly selected subsample of the dataset will come up to be imbalanced throughout the process. We made sure to preserve this balance everywhere it was possible.

Every picture was preprocessed in the same way. Specifically, each image was 1) converted to RGB, 2) resized to the shape (299,299,3), and 3) had all its pixels normalized.

As Kaggle offers a limited amount of memory, it was not feasible to process all data at once, even for the relatively small number of pictures. Therefore the custom data generator was used. It generates blocks of pictures by selected batch size as long as there is enough data for a full batch. After each epoch, it shuffles data (if such option is preferred) so that batches for each epoch never consist of the same pictures.

Training And Evaluation Process

We trained the model and checked its performance on the validation set after every epoch. We also made use of several callbacks for different purposes:

- Tensorboard callback to save details of different pieces of training and evaluation metrics while performing hyperparameter tuning,
- EarlyStopping callback to avoid overfitting and long unnecessary training,
- ReduceOnPlateau to lower learning rate when model stopped improving,
- ModelCheckpoint to save the best weights achieved during training, and a custom
- TimerCallback that stops the training and saves the weights when the time limit set for the particular training exceeded.

After receiving several arguably the best models of every type, we evaluated them on 30000 pictures, 10000 pictures per class, which were not used during training, evaluation, and testing processes for any model. After obtaining numbers of the metrics for each model, we analyze them by viewing the top 100 and the bottom 100 predicted pictures, sorted by the confidence of predictions.

0.5.2 Experimental results

First of all, we performed transfer learning with fine-tuning of the InceptionV3 model. One of the main issues was that the model started overfitting only after several epochs. For this reason, the dropout of at least 30% was an absolute necessity, as well as not bigger than $1 \cdot 10^{-2}$ learning rate. The top layers converged to more or less 85% accuracy by approximately 3-7 epochs, and after unfreezing additional layers, it converged in 10-20 more epochs depending on a particular model. The best accuracy we could achieve on the test set was 90% by several models. Their losses and confusion matrices looked alike as well.

The information about the model we considered to be the best is provided below. We trained its last layers for seven epochs and afterwards fine-tuned the model for 20 more epochs. We used the same learning rate and optimizer for both cases but reduced the learning rate manually a little bit for the last few epochs.

Hyperparameters	Values
Learning Rate	$1 \cdot 10^{-4}$
Optimizer	SGD with 0.9 momentum
Dropout	0.3
Dense layer	1024 with relu activation

TABLE 5: Model Characteristics # 1

- Model Loss: 0.267
- Model Accuracy: 0.901

t\p	selfie	person	random
selfie	0.967	0.03	0.003
person	0.035	0.864	0.101
random	0.003	0.129	0.867

TABLE 6: Confusion matrix # 1

From the normalized confusion matrix[6], we can notice that selfie class has the highest recall among other classes. Also, the model tends to misclassify random and person classes more often. It is partially true because the labels for these two classes are not 100% right. Moreover, the matrix is almost symmetric. Selfie is more likely to be confused with a person and less likely with something random.

By analyzing visual results, we highlighted tendencies as follows:

Top uncertain predictions

All classes are more or less represented among these pictures. Almost all predictions made about pictures' classes can be understood and are not entirely random. In this subset of bottom predictions, many photos were not of the best quality.

Not certain about selfies:

- of men, boys
- with animals or some objects (can be interpreted as other classes)
- when a person is strongly on one side (it seems like the model will be more likely to assume that such a picture is a selfie if a person is on the right)

The model tends to misinterpret as selfies:

- padded pictures of separate people, portraits in frames
- simple portraits
- pictures with certain people holding a camera or phone (even if there is no mirror)

- babies(one or two examples)

Other characteristics:

- people with strange outwears classified as random
- pictures in a grayscale
- outfits may be classified as a person
- naked bodies even though classified correctly
- pictures with persons and any text

See the example of bottom 50 predicted pictures[4].

Top confident predictions

All pictures in this subset were selfies. The subset consists of women by 90%, and 10% of pictures are men. All selfies are with a big face, some of them are padded and with filters applied. The model seemingly learned head positioning patterns of selfies. Pictures in the mirror are also classified as selfies with confidence.

As we could see from the top predictions, the model is the most certain about predicting selfies, which is good if we are willing to detect them. We assume that the model identifies some specific patterns for class types. Thus perhaps the model learned what typical prerequisites for a picture to be selfie are as well.

See the example of top 50 predicted pictures[5].

We performed full model training on InceptionV3 afterwards. As a result, we were also able to achieve competitive accuracy in comparison to the previous model. Although the maximum accuracy was equal to 88%, we interpret it as a good result. The best model is described below.

It trained for 15 epochs with the ReduceOnPlateau callback, which was reducing the learning rate by the factor of 0.9 every two epochs when validation loss stopped improving.

Hyperparameters	Values
Learning Rate	$1 \cdot 10^{-3}$
Optimizer	Adam
Dropout	0.4

TABLE 7: Model Characteristics # 2

- Model Loss: 0.289
- Model Accuracy: 0.889

t\p	selfie	person	random
selfie	0.99	0.009	0.001
person	0.001	0.875	0.125
random	0.000	0.198	0.802

TABLE 8: Confusion matrix # 2

From the confusion matrix[8], we observed that this model predicts selfies even better but also makes more significant mistakes in predicting random pictures as it seems to be skewed to predicting images as a person more often.

Let's move to the picturesque representation of the predictions.

Top uncertain predictions

A large number of pictures seem to be really randomly predicted without much logic. Pictures of the random category dominate over others. For example, some pictures, classified as the person class, have nothing in common with a human being. There are many pictures of art, nature, artificial images, and ones with many people. Luckily there are not so many selfies among these most uncertain pictures. Furthermore, the model faces difficulties when there is a lot of people in one picture, and they are

- situated very closely,
- are wearing strange outfits,
- are not in the typical body poses.

The model thinks that these pictures are random. It seems like the features are more shallow than in the previous model. Presumedly it is because the previous model still remembered complex features from the ImageNet.

See the example of bottom 50 predicted pictures[6].

Top confident predictions

This subset entirely consists of selfies. Furthermore, it seems to be diverse. It indeed contains more men than in the previous case.

See the example of top 50 predicted pictures[7].

Till this moment, we worked with the InceptionV3 model and tried to understand whether useful features can be extracted from the available data. As we could see, the model which "saw" the ImageNet before was able to extract more specific features than the model trained only on our dataset. But in some cases, this information may be excessive and lead to unwanted results. For example, selfies with cameras or phones made in the mirrors or more demonstrative - selfies with animals. There are many thousands of animal pictures in the ImageNet. Thus, we suppose that the model could pay too much attention to details connected to animals, which are not significant in this case. On the other hand, the model trained from scratch showed more random (less logical) results in the worst cases. Still, it detected selfies even better than the first model.

The last part of the experiments was made with the use of EfficientNet. At first, we tested how the performance changes on different EfficientNet models, starting with the backbone EfficientNetB0 and ending up with EfficientNetB3. With that particular setting we had for the test runs, we didn't notice any significant differences in performances in favor of switching to a lighter model; thus, we decided to continue working on the EfficientNetB3 model.

Here are the details of the model, which showed the best results.

- Model Loss: 0.284
- Model Accuracy: 0.879

Hyperparameters	Values
Learning Rate	$1 \cdot 10^{-3}$
Optimizer	Adam with 0.9 momentum
Dropout	0.2
Other	

TABLE 9: Model Characteristics # 3

t\p	selfie	person	random
selfie	0.996	0.004	0.000
person	0.001	0.851	0.147
random	0.001	0.209	0.791

TABLE 10: Confusion matrix # 3

This model has the worst results in classifying persons and random images. It is similar to the previous model as it has more troubles with random data. Despite this, it makes the least errors about selfie class.

Top uncertain predictions The first thing we noticed is that most bad pictures are dark, saturation is very low. In particular, the model faces difficulties with black and white selfies and black and white pictures overall. It is not confident in detecting graphical pictures and not photographic pictures. A significant amount of random pictures classified as persons catches the eye. Statues and paintings may be classified as a person too. We also noticed the same pattern as other models had. Pictures with many people in not typical outfits or doing something in diverse positions, when the background is too variegate or just there is too much of it tend to be classified as random pictures.

See the example of bottom 50 predicted pictures[8].

Top confident predictions All best-classified pictures are selfies, as in both previous cases. One thing that differs is that almost all these selfies are in vibrant colors or with filters. Perhaps the model learned stylistic features that you can generally find to be at selfies. Thus the type of selfies in such a color scheme is predicted with more confidence.

We may assume that the last model is picking up not necessarily the head positioning or other relevant content, but also anything related to image manipulation. As top classified pictures are very bright and bottom classified pictures are mostly dull, we suppose that the model actually is paying much more attention to color distribution than needed. If the model is overtrained on color distribution rather than content, it may work for the current data; still when the model sees selfies with normal color distribution, there is a bigger risk of misclassification. This fact makes this model not so attractive.

See the example of top 50 predicted pictures[9].

Model	Train Loss	Train Acc	Test Loss	Test Acc
Model 1	0.2017	0.9246	0.2708	0.9007
Model 2	0.1993	0.9191	0.3463	0.8523
Model 3	0.2499	0.8931	0.3463	0.8661

TABLE 11: Models' Loss And Accuracy Comparison

This is the additional table[11] of the statistics gathered during the training process.

0.6 Bias Analysis

0.6.1 Experimental Setup

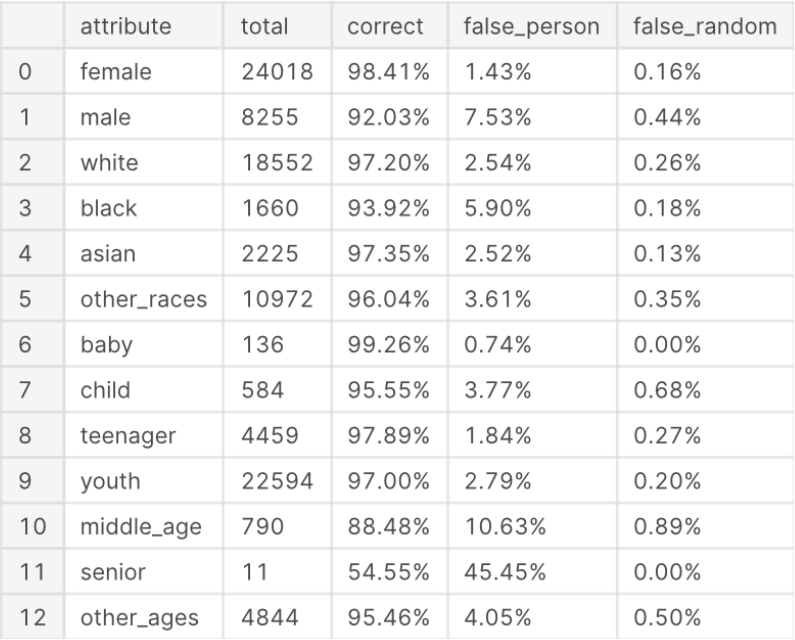
We performed the analysis of bias on 33467 selfies that were not used during the training process. For each of the three models, we explore the attributes which relate to demographics. For each attribute, we output the total amount of pictures that have this attribute, which percent of images are predicted correctly, which percent is falsely predicted as a person and as random.

We also describe the observations about demographics we made from visualizations mentioned earlier in this work. These visualizations of the top pictures predicted with the most confidence and bottom pictures predicted with the least confidence may help us to make assumptions about the demographical features of both cases if there are some.

We do not have enough pictures of each attribute to state about any results with 100% confidence, but still, we can assume some tendencies from the information we have.

0.6.2 Experimental Results

In the tables below results for model #1[6], model #2[8], and for model #3[10] in this order.

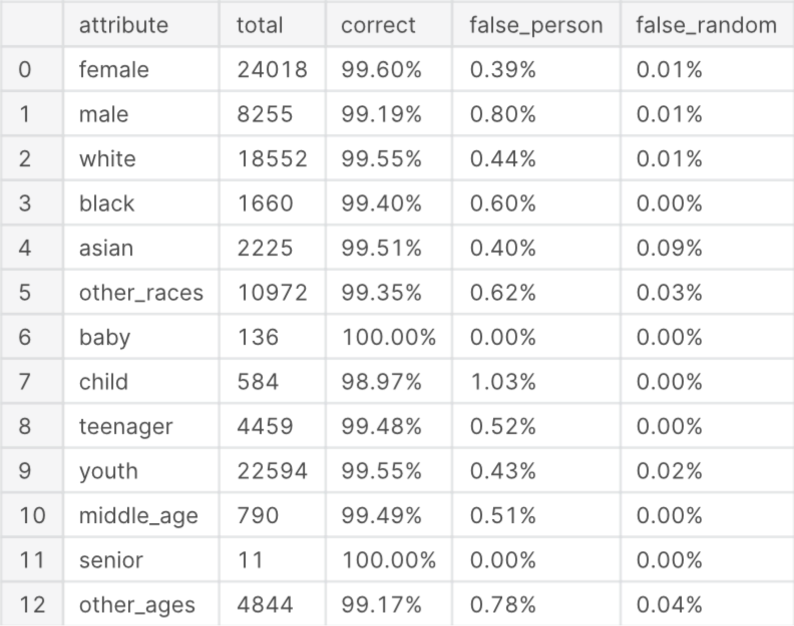


	attribute	total	correct	false_person	false_random
0	female	24018	98.41%	1.43%	0.16%
1	male	8255	92.03%	7.53%	0.44%
2	white	18552	97.20%	2.54%	0.26%
3	black	1660	93.92%	5.90%	0.18%
4	asian	2225	97.35%	2.52%	0.13%
5	other_races	10972	96.04%	3.61%	0.35%
6	baby	136	99.26%	0.74%	0.00%
7	child	584	95.55%	3.77%	0.68%
8	teenager	4459	97.89%	1.84%	0.27%
9	youth	22594	97.00%	2.79%	0.20%
10	middle_age	790	88.48%	10.63%	0.89%
11	senior	11	54.55%	45.45%	0.00%
12	other_ages	4844	95.46%	4.05%	0.50%

FIGURE 1: Bias Run #1

The model which was trained with transfer learning shows the most biased result. It has more trouble with classifying pictures with men, dark-skinned people, children, and middle-aged people, especially seniors.

There were many pictures with men and boys on the photos, which were complicated for the model to predict. On the other hand, images from top predictions consisted mostly of white-skin women. The ratio between women and men is approximately 9:1, for top-predicted pictures.

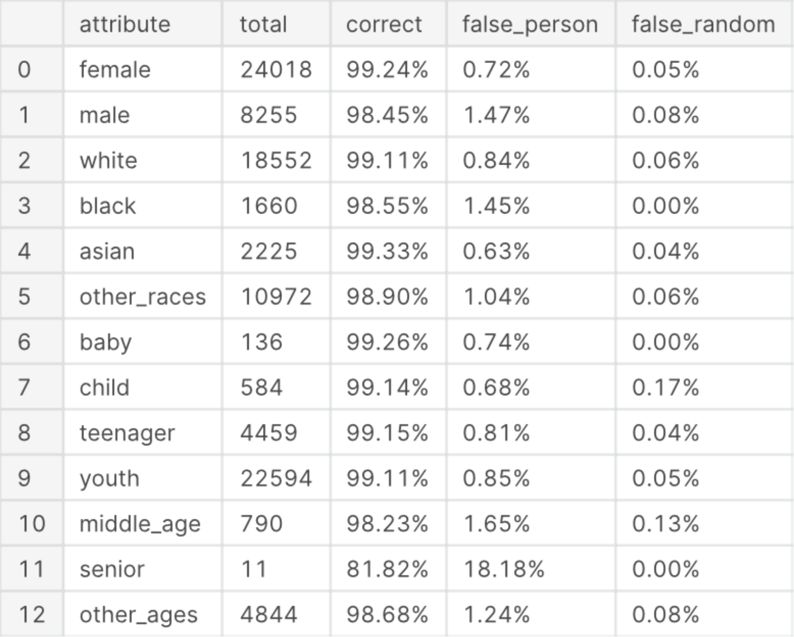


	attribute	total	correct	false_person	false_random
0	female	24018	99.60%	0.39%	0.01%
1	male	8255	99.19%	0.80%	0.01%
2	white	18552	99.55%	0.44%	0.01%
3	black	1660	99.40%	0.60%	0.00%
4	asian	2225	99.51%	0.40%	0.09%
5	other_races	10972	99.35%	0.62%	0.03%
6	baby	136	100.00%	0.00%	0.00%
7	child	584	98.97%	1.03%	0.00%
8	teenager	4459	99.48%	0.52%	0.00%
9	youth	22594	99.55%	0.43%	0.02%
10	middle_age	790	99.49%	0.51%	0.00%
11	senior	11	100.00%	0.00%	0.00%
12	other_ages	4844	99.17%	0.78%	0.04%

FIGURE 2: Bias Run #2

From bias perspective this result is the best one. One percent is the biggest number among percentages of not correctly classified attributes. And even if we compare ratios of polar attributes for the first model and this one (e.g. female/male or white/black) we achieve smaller numbers.

Top predicted pictures seem to contain people of diverse races and skin colors. The ratio between women and men is approximately 9:2, which is a better result than for the previous model.



	attribute	total	correct	false_person	false_random
0	female	24018	99.24%	0.72%	0.05%
1	male	8255	98.45%	1.47%	0.08%
2	white	18552	99.11%	0.84%	0.06%
3	black	1660	98.55%	1.45%	0.00%
4	asian	2225	99.33%	0.63%	0.04%
5	other_races	10972	98.90%	1.04%	0.06%
6	baby	136	99.26%	0.74%	0.00%
7	child	584	99.14%	0.68%	0.17%
8	teenager	4459	99.15%	0.81%	0.04%
9	youth	22594	99.11%	0.85%	0.05%
10	middle_age	790	98.23%	1.65%	0.13%
11	senior	11	81.82%	18.18%	0.00%
12	other_ages	4844	98.68%	1.24%	0.08%

FIGURE 3: Bias Run #3

The numbers and their ratios for the fully-trained EfficientNetB3 model are better than for the first model but worse than we gained with the fully-trained InceptionV3 model.

The ratio of men and women in most certainly predicted pictures is the same as for the first model. Nothing special could be singled out except that the were several girls wearing hijab in the top predictions.

As we expected, not all demographic types are recognized equally well. Models tend to misclassify

- men,
- people with dark skin or
- with other appearances which are not that common,
- people of not typical ages to take a selfie.

These attributes are usually the ones that are not represented good enough in the data, and we can observe the negative correlation between the number of pictures per attribute and the percentage of badly predicted pictures. By comparing the results of the model which "saw" pre-trained weights before and models which were trained entirely from scratch, we can assume that in the first case, the model is carrying in some part of bias from the ImageNet. Given the complex features the network is trained on, it is more able to distinguish between three different classes, as could be seen in the previous section. However, this happens at the cost of the model's fairness. For the selfie detection task, such tendency is inappropriate as it worsens the desired performance.

0.7 Conclusion

0.7.1 Summary

In this work, we investigated how different models performed selfie detection and analyzed their demographic bias. Although selfie detection has not been widely tested before, it is a fairly feasible task with the help of state-of-the-art models that are now available.

Because of the thickness and complexity of the pre-trained classifier, transfer learning shows better results across different classes. However, when we train models from scratch, we obtain better accuracy for the class that we are interested in most.

Since the problem setting was to reduce false-negative results for selfies in the first place, it would also be justified to use a network trained from scratch. Especially because we saw evidence that by training models from scratch, we could eliminate some part of the bias, at least the one inherited from the transfer model.

Finally, although the EfficientNet is a new promising architecture that out-performs other networks, for this task, it might not be the best choice because from our experience, it is faking up stylistic features. Speaking of the style of the images, especially the color, it can work for the control dataset, but it might create errors while the model is tested in the wild.

0.7.2 Further work

The future work on the selfie detection may include improvements of all steps we made for this thesis. We plan to realize and present a part of it at the final presentation of the thesis, which will be held in June.

The ways to improve this work include:

- perform thorough bias analysis for the person class on the dataset with demographics we created
- test selfie detection in the wild on the data from Wiki Loves Africa
- extend and improve both datasets by adding new pictures and metadata, make them cleaner
- train the classifier to be more unbiased
- work on distinguishing real people and works of arts
- try other models, other approaches, and any applicable manipulations to improve current results
- consider ways to make this work more useful for Wikimedia Commons

.1 Examples Of Visualization

.1.1 Transfer Learned InceptionV3 Model

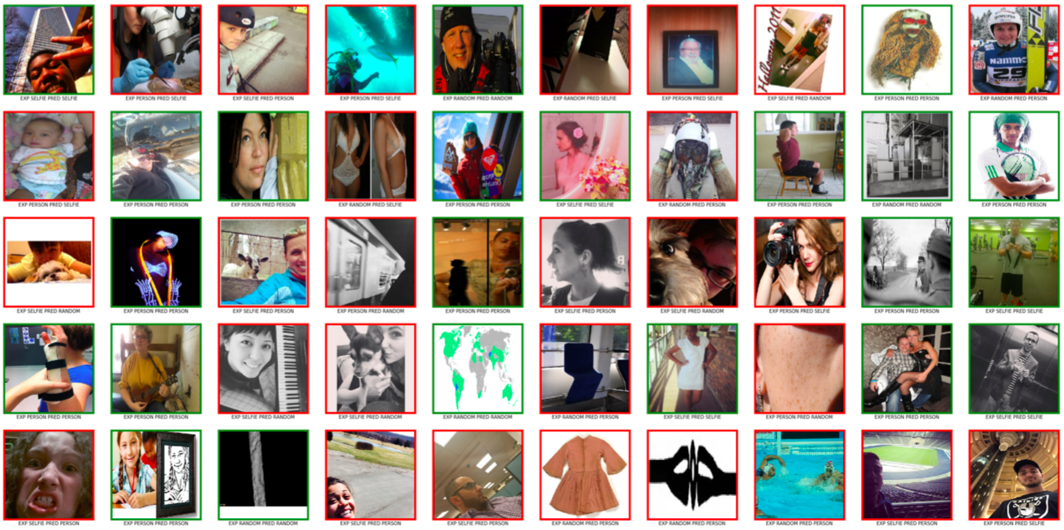


FIGURE 4: Bottom 50 Predicted Pictures #1

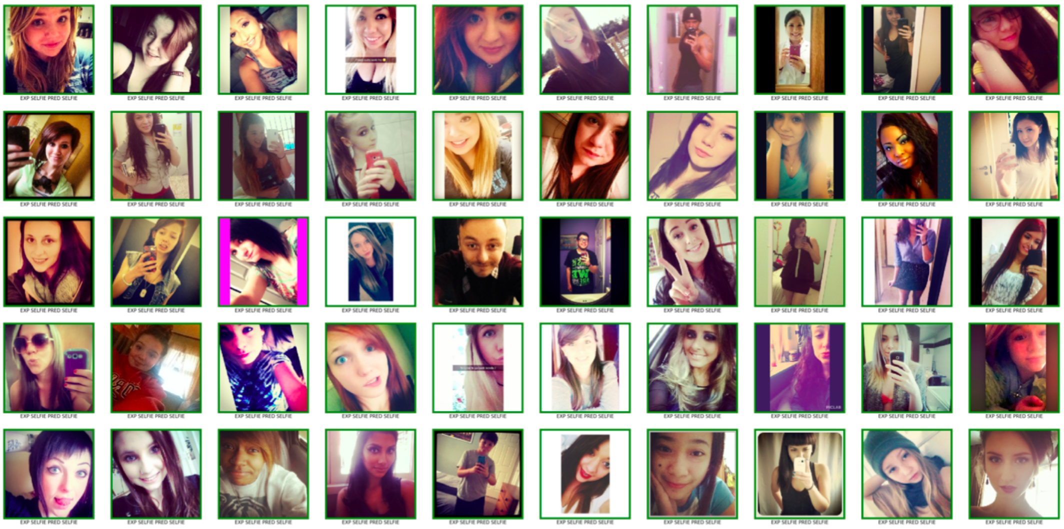


FIGURE 5: Top 50 Predicted Pictures #1

.1.2 Fully Trained InceptionV3 Model

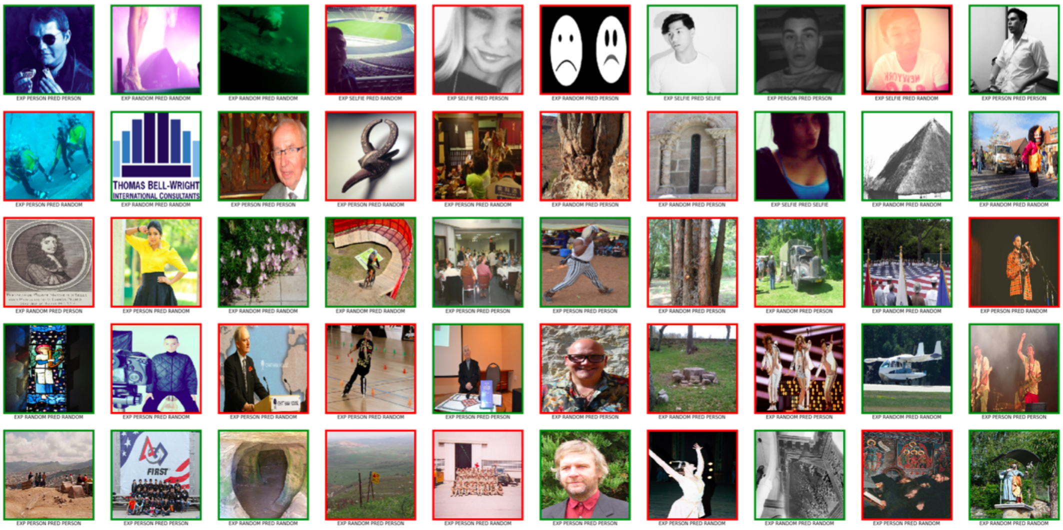


FIGURE 6: Bottom 50 Predicted Pictures #2

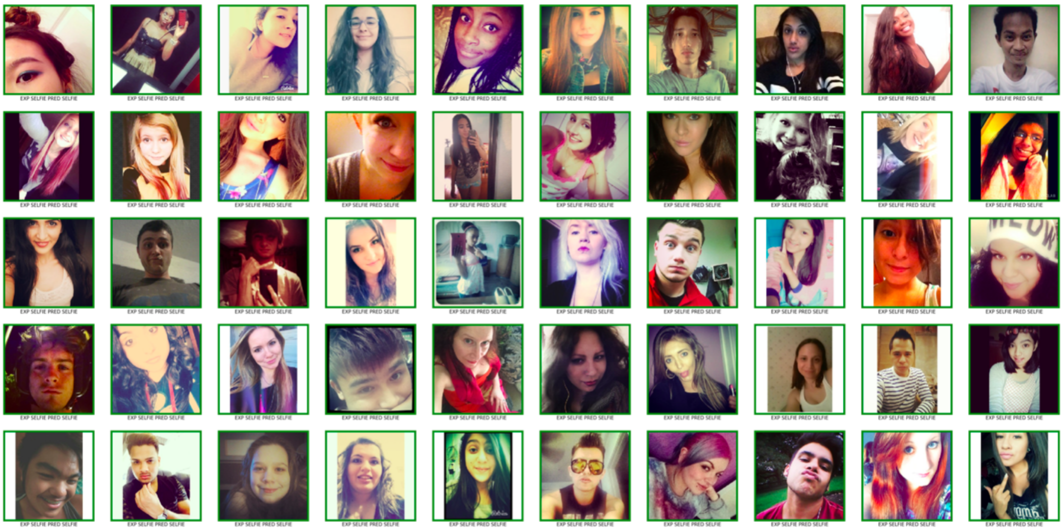


FIGURE 7: Top 50 Predicted Pictures #2

.1.3 Fully Trained EfficientNetB3 Model

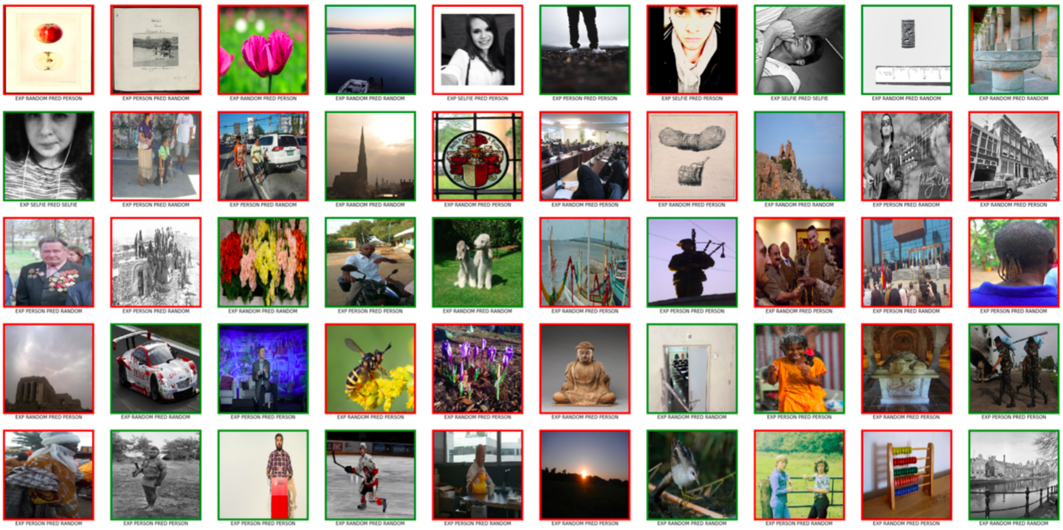


FIGURE 8: Bottom 50 Predicted Pictures #3

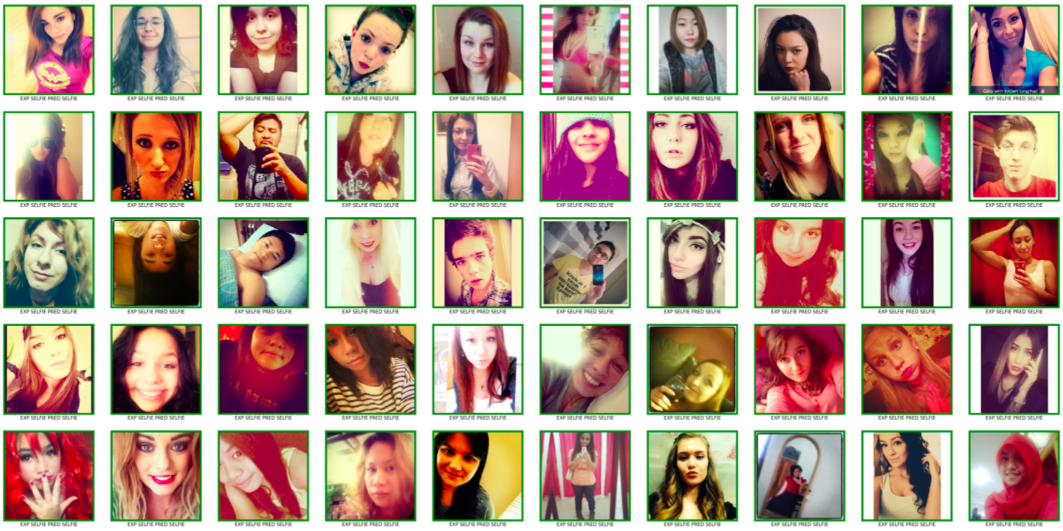


FIGURE 9: Top 50 Predicted Pictures #3

Bibliography

[1] Y Annadani et al. “Selfie Detection by Synergy-Constraint Based Convolutional Neural Network”. In: *2016 12th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)*. ieeexplore.ieee.org, Nov. 2016, pp. 335–342. URL: http://dx.doi.org/10.1109/SITIS.2016.61.

[2] BRADSKI and G. “The OpenCV library”. In: *Dr Dobb's J. Software Tools* 25 (2000), pp. 120–125. URL: https://ci.nii.ac.jp/naid/10028167478/.

[3] Tianlang Chen, Yuxiao Chen, and Jiebo Luo. “A Selfie is Worth a Thousand Words: Mining Personal Patterns behind User Selfie-posting Behaviours”. In: *Proceedings of the 26th International Conference on World Wide Web Companion*. WWW '17 Companion. Perth, Australia: International World Wide Web Conferences Steering Committee, Apr. 2017, pp. 23–31. URL: https://doi.org/10.1145/3041021.3054142.

[4] J Deeb-Swihart et al. “Selfie-presentation in everyday life: A large-scale characterization of selfie contexts on instagram”. In: *Eleventh International AAAI* (2017). URL: https://www.aaai.org/ocs/index.php/ICWSM/ICWSM17/paper/viewPaper/15692.

[5] J Deng et al. “ImageNet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. ieeexplore.ieee.org, June 2009, pp. 248–255. URL: http://dx.doi.org/10.1109/CVPR.2009.5206848.

[6] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. en. MIT Press, Nov. 2016. URL: https://play.google.com/store/books/details?id=omivDQAAQBAJ.

[7] Mahdi M Kalayeh et al. “How to take a good selfie?” In: *Proceedings of the 23rd ACM international conference on Multimedia*. dl.acm.org, 2015, pp. 923–926. URL: https://dl.acm.org/doi/abs/10.1145/2733373.2806365.

[8] Kimmo Kärkkäinen and Jungseock Joo. “FairFace: Face Attribute Dataset for Balanced Race, Gender, and Age”. In: (Aug. 2019). arXiv: 1908.04913 [cs.CV]. URL: http://arxiv.org/abs/1908.04913.

[9] Hao Li et al. “Rethinking the Hyperparameters for Fine-tuning”. In: (Feb. 2020). arXiv: 2002.11770 [cs.CV]. URL: http://arxiv.org/abs/2002.11770.

[10] V S Priyanka, B Hussain, and R P Aneesh. “Genuine Selfie detection Algorithm for Social media Using Image Quality Measures”. In: *2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)*. ieeexplore.ieee.org, Dec. 2018, pp. 1–6. URL: http://dx.doi.org/10.1109/ICCSDET.2018.8821075.

[11] Lin Qiu et al. “What does your selfie say about you?” In: *Comput. Human Behav.* 52 (Nov. 2015), pp. 443–449. URL: http://www.sciencedirect.com/science/article/pii/S0747563215004720.

[12] Ajita Rattani, Reza Derakhshani, and Arun Ross. *Selfie Biometrics: Advances and Challenges*. en. Springer Nature, Sept. 2019. URL: https://play.google.com/store/books/details?id=bw2xDwAAQBAJ.

[13] C Szegedy et al. "Inception-v4, inception-resnet and the impact of residual connections on learning". In: *Thirty-first AAAI conference on* (2017). URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/viewPaper/14806.

[14] Christian Szegedy et al. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. cv-foundation.org, 2015, pp. 1–9. URL: https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Szegedy_Going_Deeper_With_2015_CVPR_paper.pdf.

[15] Christian Szegedy et al. "Rethinking the Inception Architecture for Computer Vision". In: (Dec. 2015). arXiv: 1512.00567 [cs.CV]. URL: http://arxiv.org/abs/1512.00567.

[16] Mingxing Tan and Quoc V Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: (May 2019). arXiv: 1905.11946 [cs.LG]. URL: http://arxiv.org/abs/1905.11946.

[17] Jason Yosinski et al. "How transferable are features in deep neural networks?" In: *Advances in Neural Information Processing Systems 27*. Ed. by Z Ghahramani et al. Curran Associates, Inc., 2014, pp. 3320–3328. URL: http://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks.pdf.