NSFW Classifier.md

# NSFW Classifier

## Problem Statement

To make browsing web and popular social media platforms like Facebook, Instagram, Twitter, Tumblr more safer for children by curbing down on NSFW content present in the form of Images & Videos.

## Business Problem

Of all the content present on the Internet, 30% contains pornography and much more contains NSFW content. With abundance of such content, it is required to build a tool which can filter floating images and videos so as build a safer community and web experience.

## Data Overview

We made use of NSFW Data Scraper in order to construct the entire dataset. This repository comprises a set of scripts that allows for an automatic collection of tens of thousands of images for the following (loosely defined) categories to be later used for training an image classifier:

- **porn** - pornography images
- **hentai** - hentai images, but also includes pornographic drawings
- **sexy** - sexually explicit images, but not pornography. Think nude photos, playboy, bikini, etc.
- **neutral** - safe for work neutral images of everyday things and people
- **drawings** - safe for work drawings (including anime)

After collecting all the images and removing the corrupted ones, we ended up with roughly 74k total images which accounted for approximately 34GBs of disk space. For the first experiment, we made use of a subset of this entire dataset which contained roughly 30k images (20k for training and 10k for testing)and occupied 12GBs of disk space. The following table depicts the data structure:

| Categories | Train | Test |
|---|---|---|
| Drawing | 2000 | 2000 |
| Hentai | 2000 | 2000 |
| Neutral | 8000 | 2000 |
| Porn | 3000 | 2000 |
| Sexy | 5000 | 2000 |
| **Total** | **20000** | **10000** |

## Business Objectives & Constraints

- **Low Latency Requirement:** While browsing web, we wanted the model to make inference as quickly as possible so that there isn't any lag in content getting identified which might ultimately lead to bad user experience.
- **Confidence Probability:** Model should be able to predict confidence probability scores for each prediction being made.
- **High Accuracy:** Model should be able to identify NFSW images & ignore the safe ones with high accuracy.

## Performance Metrics

- High Precision & High Recall
- Confusion Matrix

## Techniques

We followed Supervised Learning paradigm from the realm of Machine Learning. As the data we worked upon was unstructured in nature i.e. images/videos, it made more sense to go with Deep Learning based models like Convolutional Neural Nets. Although, we had good amount of data to work with, but knowing that the ImageNet contains some NSFW images, we used pretrained architectures and later fine-tuned over images in our dataset. We made use of the following models pretrained on ImageNet:

- MobileNetV2 Smaller Weights, Faster Loading & Inference, Decently Accurate
- NASMobileNet Smaller Weights, Faster Loading & Inference, Decently Accurate
- EfficientNetB0 Comparably Larger Weights, Comparably Slower Loading, Better Accuracy

## Experimentation & Validation

We trained aforementioned architectures over the training dataset spanning over 5 classes and comprising of roughly 20k images. Following are the transformations performed on the training images:

```
rescale=1./255,
rotation_range=30,
width_shift_range=0.2,
height_shift_range=0.2,
shear_range=0.2,
zoom_range=0.2,
channel_shift_range=20,
horizontal_flip=True
```

Following are model setup details:

```
image_size=224
epochs=100
steps=500
```

Following is the model-head added to the aforementioned base architectures (MobileNet/NASMobileNet/EfficientNetB0):

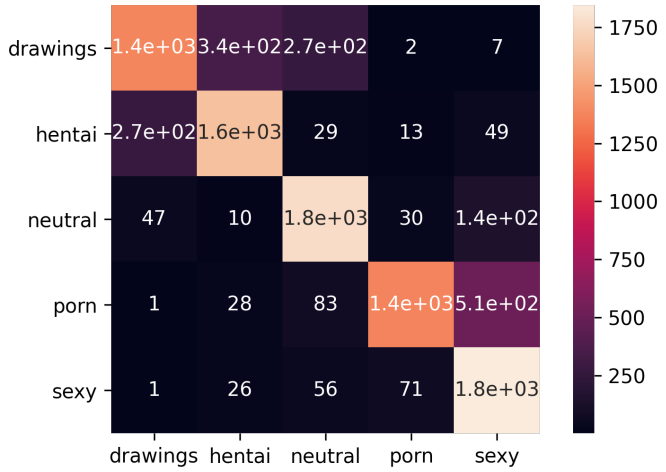
```
base model
⬇
AveragePooling2D with pooling size (7, 7)
⬇
Flatten
⬇
Dense(32) with ReLU
⬇
Batch Normalization
⬇
Dropout (0.5)
⬇
Dense(5) with SoftMax
```

Later, we validated our model using images in our testing set. We didn't perform any transformation other than normalizing/rescaling them.

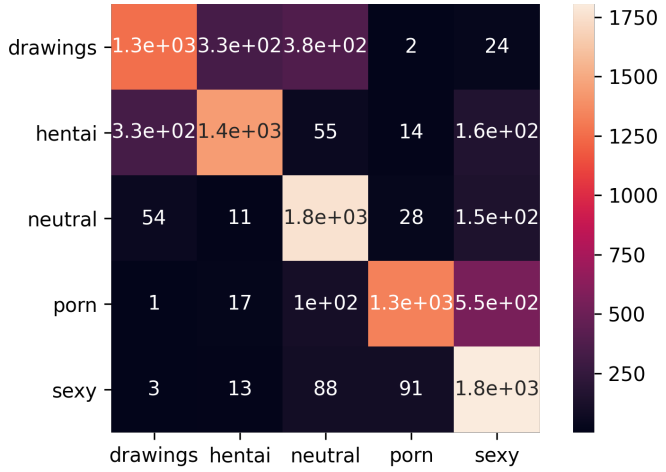
```
rescale=1./255
```

## Results

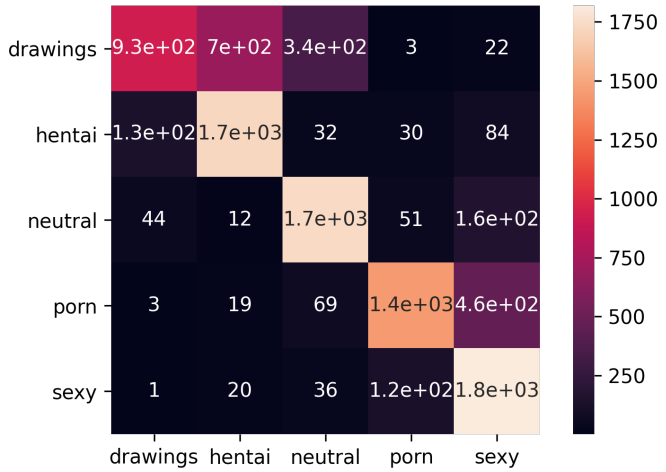
- **Confusion Matrix**
    - **MobileNetV2**




- **NASMobileNet**




- **EfficientNet**




- **Classification Report**
  - **MobileNetV2**

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| **drawings** | 0.8115091015854374 | 0.691 | 0.7464218201458277 | 2000.0 |

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| **hentai** | 0.800880626223092 | 0.8185 | 0.8095944609297724 | 2000.0 |
| **neutral** | 0.803352967829633 | 0.8865 | 0.8428809127644401 | 2000.0 |
| **porn** | 0.9222520107238605 | 0.688 | 0.7880870561282932 | 2000.0 |
| **sexy** | 0.7227877838684417 | 0.923 | 0.8107158541941152 | 2000.0 |
| **accuracy** | | | 0.8014 | 10000.0 |
| **macro avg** | 0.8121564980460928 | 0.8013999999999999 | 0.7995400208324897 | 10000.0 |
| **weighted avg** | 0.8121564980460929 | 0.8014 | 0.7995400208324897 | 10000.0 |

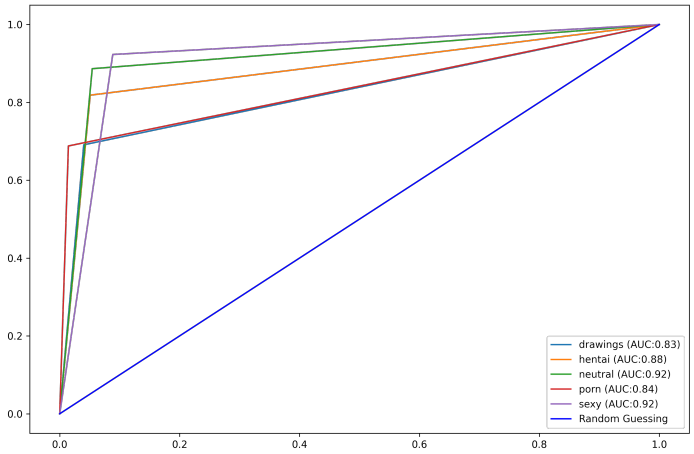
- **NASMobileNet**

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| **drawings** | 0.7647058823529411 | 0.6305 | 0.6911482597972046 | 2000.0 |
| **hentai** | 0.7942794279427943 | 0.722 | 0.7564169722367732 | 2000.0 |
| **neutral** | 0.7376362112321878 | 0.88 | 0.8025535795713634 | 2000.0 |
| **porn** | 0.9076607387140903 | 0.6635 | 0.7666088965915655 | 2000.0 |
| **sexy** | 0.6722532588454376 | 0.9025 | 0.7705442902881536 | 2000.0 |
| **accuracy** | | | 0.7597 | 10000.0 |
| **macro avg** | 0.7753071038174901 | 0.7596999999999999 | 0.7574543996970121 | 10000.0 |
| **weighted avg** | 0.7753071038174904 | 0.7597 | 0.7574543996970121 | 10000.0 |

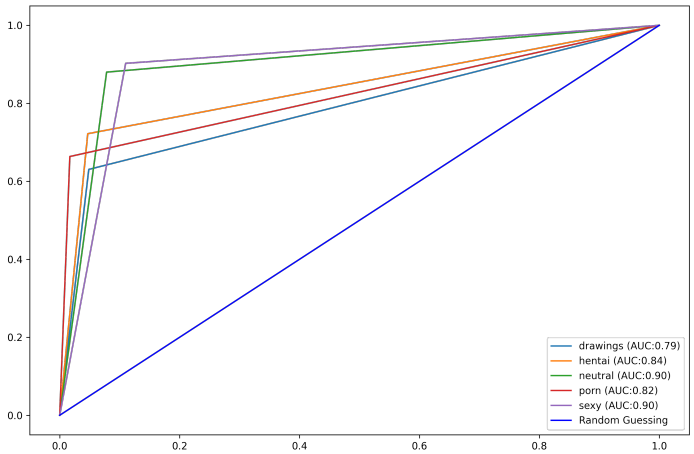
- **EfficientNet**

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| **drawings** | 0.8395285584768812 | 0.463 | 0.5968417660328714 | 2000.0 |
| **hentai** | 0.6955645161290323 | 0.8625 | 0.7700892857142858 | 2000.0 |
| **neutral** | 0.7825890843482183 | 0.8675 | 0.8228598529760494 | 2000.0 |
| **porn** | 0.8741681790683605 | 0.7225 | 0.791130577607446 | 2000.0 |
| **sexy** | 0.7141735374950923 | 0.9095 | 0.8000879700901694 | 2000.0 |
| **accuracy** | | | 0.765 | 10000.0 |
| **macro avg** | 0.7812047751035169 | 0.765 | 0.7562018904841645 | 10000.0 |
| **weighted avg** | 0.7812047751035169 | 0.765 | 0.7562018904841643 | 10000.0 |

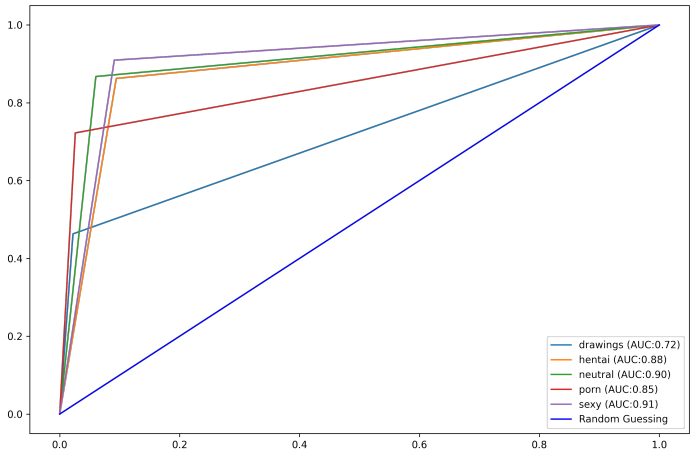
- **ROC-AUC Curve**
  - **MobileNetV2**




- **NASMobileNet**




- **EfficientNet**




# Pitfalls

Although, the model gives good enough results but still suffers from few issues.

-

Sometimes the model gets confused between sexy and porn images. There are images which could be either of them yet one of them is predicted and that too with low confidence scores. Possible reason could be due to the training dataset being noisy, the model trained upon it, is sometimes unsure and unconfident of its predictions.

- The model is not able to classify those porn images as porn which contain male genitalia predominantly. Possible reason could be that the majority of the porn images are female dominant leading to the males not being centric for featurization to take advantage of.
- Sometimes models predict the input image to be either hentai or drawing interchangeably. Possible reasons could be due to low representation and not so much difference between hentai and drawing categories.

## Solutions

We believe following could help us mitigate some of the above mentioned issues:

- Less noisy dataset.
- More representation for hentai and drawing categories.
- Using a separate classifier model attributed to label an image to be containing female and male body parts for better combined confidence score.