Table of Contents

Candidate Details

Name: Harshinee Sriram

Email: sriramharshinee@gmail.com | hsriram@cs.ubc.ca

Contact number: +91 8369516382

Website/Resume: http://harshineesriram.github.io/

LinkedIn: https://www.linkedin.com/in/harshineesriram/

GitHub: https://github.com/HarshineeSriram

Location: India (UTC +5:30) and, if my study permit is approved, Canada (UTC -7:00)

Typical working hours: 12 PM to 8 PM at (UTC +5:30), and, if my study permit for Canada is approved, (UTC -7:00)

Synopsis of the project

Wikimedia Commons is an online repository of free-use images, sounds, other media, and JSON files. Anyone can upload media to the Commons portal. The uploads are moderated by members and volunteers of the foundation manually. This project aims to build a classifier that can flag NSFW images/media for review. Upon successful completion of this internship, the intern would have designed, implemented and tested a machine learning model that would be able to classify image and video media as SFW or NSFW with a high accuracy. They would also be given the chance to deploy the model to Wikimedia test and production servers. Further, they would build a data processing pipeline and an API for the model.

Since this is a scratch project, applicants are required to do some research initially. A basic comparison of the existing NSFW classifiers along with their computational requirements is required. All applicants are expected to read various research papers and draw comparisons between them. They are expected to come up with a report detailing their research, the various options that can be used to implement the model and what they are proposing to do if they are selected. This report should also detail implementational methods and procedures.

Mentors: @abbasidaniyal and @chtnnh

Expected Internship Tasks

1) **Developing an image classifier for NSFW images [https://phabricator.wikimedia.org/T264049]:**

   Design and implement a machine learning model which classifies images uploaded to Wikimedia Commons as Safe for Work (SFW) or Not Safe for Work (NSFW). Frameworks like Tensorflow / PyTorch can be used. The model should not be very computationally intensive and it should also be well tested.

2) **Creating an API for NSFW classification [https://phabricator.wikimedia.org/T264052]:**

   A NSFW classifier model that takes an image as input and classifies it as SFW or NSFW.
   Model can be served using something like Tensorflow Serving (if built with tensorflow). This task requires building a RESTful API for the machine learning model. The API can be built using frameworks like Flask / Django / FastAPI etc. The API will contain a simple endpoint that receives an image, passes it through the model and returns the NSFW classification result as well as the confidence.

3) **Video processing module for NSFW classifier [https://phabricator.wikimedia.org/T264050]:**

   A NSFW classifier takes an image as input and classifies it as SFW or NSFW.
   This task aims at adding support for videos for the same model.

   1. Video Segmentation : Divide the video into difference scene segments using PySceneDetect or something similar.
   2. Frame Extraction : From the segmented video clips, extract indivisual frames using Decord or something similar.
   3. Feed the individual frames to the NSFW model and generate an overall score for the video.

Proposed Timeline (complete)

Since there are 3 major tasks to be completed in 12 weeks and the major task (around which the other 2 tasks revolve) is to develop a competitive model for image and video classification that can also withstand potential adversarial attacks. One additional task that goes hand-in-hand is developing an easy to understand documentation for further technology developments.

Events underlined are the default events that Outreachy will host.

Date	Task(s) Description
December 1 2020	• Blog prompt: "Introduce yourself" • Work on the dataset building along with the extent of NSFW content needed and the type of classifier to work with. • Address the video content filtration module by determining if it needs to work with the same dataset or a different one with the same framework.
December 7 2020	• Finalize on the dataset and the method of approach for image and video classification and start working on the classifier details (such as extent of classification, different categories, dataset curation) • Begin testing models and methods of approaches that would best fit the pre-determined image classifier characteristics and can be used for video classification as well.
December 8 2020	• Introduction chat
December 11 2020	• INITIAL FEEDBACK DUE
December 15 2020	• Blog prompt: "Everybody struggles"
December 22 2020	• Share something you're stuck on
December 29 2020	• Blog prompt: "Think about your audience" • Finish building the preliminary model architectures that helps with NSFW content detection (low computational power + high accuracy with low false negative rate) • Start modifying the architecture to make it better equipped for robust classification
January 5 2021	• Share why you're passionate about your project or community
January 12 2021	• Mid-point project progress blog post • MID-POINT FEEDBACK DUE
January 19 2021	• Share one "easy" project goal that took longer than expected
January 22 2021	• Finish building the classifier. • Start working on building the API
January 26 2021	• Blog prompt: "Career opportunities"

February 2 2021	• (Alums invited!) Advancing your career
February 9 2021	• No blog post - interns work on their resume • Finish building an API that works with the Wikimedia servers • Develop the necessary documentation for future improvements/for understanding the method of approach
February 16 2021	• Networking skills
February 23 2021	• Finish everything, model, API, and relevant documentation • Final project progress blog post
March 2 2021	• Internship wrap up chat • FINAL FEEDBACK DUE

Proposed Timeline (just the important parts)

Date	Task(s) Description
December 1 2020	• Work on the dataset building along with the extent of NSFW content needed and the type of classifier to work with. • Address the video content filtration module by determining if it needs to work with the same dataset or a different one with the same framework.
December 7 2020	• Finalize on the dataset and the method of approach for image and video classification and start working on the classifier details (such as extent of classification, different categories, dataset curation) • Begin testing models and methods of approaches that would best fit the pre-determined image classifier characteristics and can be used for video classification as well.
December 29 2020	• Finish building the preliminary model architectures that helps with NSFW content detection (low computational power + high accuracy with low false negative rate) • Start modifying the architecture to make it better equipped for robust classification
January 22 2021	• Finish building the classifier. • Start working on building the API
February 9 2021	• Finish building an API that works with the Wikimedia servers • Develop the necessary documentation for future improvements/for understanding the method of approach
February 23 2021	• Finish everything, model, API, and relevant documentation

Project deliverables

1. Initial Feedback: Comparison of all important classifier models to determine best model in terms of processing requirements
2. Mid-point Feedback: Finish building the early stage model and begin with modifications to better suit our inputs and the kind of information we would like to show
3. Final Feedback: Build final model, create an API for the same, deploy it, add relevant documentation for the same

Work completed till date (Project contributions)

1) Microtasks 1 and 2

I compare existing NSFW classifiers and also look up datasets that I can use. In order to identify what is NSFW, the architecture needs to know how NSFW content looks like and develop a hypothesis that approximates the actual function which determines the result. It is not feasible to develop an entire dataset from scratch with labelling, and hence, I propose an amalgamation of existing extremely relevant datasets and web scraping.

Phabricator microtask descriptions:

Link: https://github.com/HarshineeSriram/Wikimedia-NSFW-Classifier-Reports/blob/main/reports/HarshineeSriram/NSFW_Classifier_for_Wikimedia_Harshinee_Sriram.ipynb

2) Curating a preliminary dataset in order to determine what kind of an image classifier will work and to identify shortcomings

I decided to work with just one dataset for now, for trial purposes. I will add on to this dataset after testing architectures. I opted for the "NSFW Data Source URLs" developed by Data Scientist Evgeny Bazarov. This is a large and high-quality image dataset of sexually explicit images containing over 1.58 million data volumes in 159 categories.

I decided to construct a binary classifier (two outcomes: NSFW or not-NSFW).

Link: https://github.com/HarshineeSriram/Wikimedia-NSFW-Classifier-Reports/blob/main/reports/HarshineeSriram/1)%20How%20I%20curated%20a%20tentative%20dataset%20for%20finding%20the%20right%20model.ipynb

3) Applying Transfer Learning on the Preliminary Curated Dataset

I chose transfer learning because of number of training/validation samples I have. I have around 2400 training images (1200 for NSFW and 1200 for SFW), 800 validation images (400 for NSFW and 400 for SFW), and 800 test images, following the 60%-20%-20-% split.

I thought of approaching transfer learning because it is possible to get a high accuracy even with less number of examples to train/validate on. *The model I used in this example is Inception V3.*

Link: https://github.com/HarshineeSriram/Wikimedia-NSFW-Classifier-Reports/blob/main/reports/HarshineeSriram/2)%20Trying%20the%20Transfer%20Learning%20Approach%20-%20Inception%20V3.ipynb

4) Improving my dataset and further improving my hyper-parameters

➔ I add to the dataset but web scraping and NSFW art-specific dataset downloads.

➔ I probably had around 5000 images in each class at the end. I ran them through an image duplication detection script. After this, I ran the following script to make sure that all the final images are actually readable (this is because I received an error saying: cannot identify image file <_io.BytesIO object). This made me realize that if I want to increase the data in my datasets next time, I should not visit Shutterstock because almost all of their thumbnails are bytes files which are unreadable by PIL and hence cannot be trained on.

➔ This is my final dataset:

NSFW Train: 3231 images
NSFW Validation: 1076 images
NSFW Test: 1066 images

SFW Train: 3140 images
SFW Validation:1048 images
SFW Test: 1060 images

➔ I added the callbacks of "early stopping", "reduce LR", and model checkpoint (although I disabled the save_best_only for now). I also reduce the initial learning rate from 0.0001 to 0.00001. I only trained it for 20 epochs because this took me a while, but I am guessing that training for 30-40 epochs would be ideal. I was able to achieve an accuracy of 94.44% on the Validation set despite the huge increase in data.

➔ Now that I have a moderately good dataset that encompasses different types of NSFW as well as SFW media, I can use this for comparable performance. Next, I will be implementing the MobileNet architecture, which was my originally proposed model due to a less amount of hyper-parameters to regulate as well as due to the light-weight nature of the model (it's only a couple of MB huge).

➔ As of now, I am still finding solutions on the following drawbacks:
- Not enough training data for POC: the majority of the NSFW images that were scrapped are of Caucasian people
- Building a model to detect NSFW content in videos

Link: https://github.com/HarshineeSriram/Wikimedia-NSFW-Classifier-Reports/blob/main/reports/HarshineeSriram/3)%20Hyperparameter%20and%20Dataset%20updates%20for%20notebook%202.ipynb

5) Trying to improve the InceptionV3 model

This notebook builds on the previously modified InceptionV3 model, experimenting with new layers. I tried changing the architecture more this time. This includes increasing the number of trainable neurons and increasing the dropout layers. In this architecture, I add 3 fully trainable neuron layers and 2 dropout layers to the output layer. Even though this prevents overfitting to an extent, the final performance is similar to the previous architecture.

Link: https://github.com/HarshineeSriram/Wikimedia-NSFW-Classifier-Reports/blob/main/reports/HarshineeSriram/4)%20Architectural%20changes%20for%20Transfer%20Learning%20-%20InceptionV3.ipynb

6) Implementing the MobileNet architecture

In this notebook, I test out the model that I believe would be a good choice for the final preliminary model because of its characteristics of being computationally relatively inexpensive and low amount of hyper-parameter tuning.

Even though this model is smaller than the InceptionV3, it already performs better with lower validation loss and higher validation accuracy (with a max val_accuracy of 96.62% after only 7 epochs) despite similar kinds of layers

applied to both (this and the InceptionV3). Additionally, because this is more robust, the time to run each epoch was also lesser than the InceptionV3.

Link: https://github.com/HarshineeSriram/Wikimedia-NSFW-Classifier-Reports/blob/main/reports/HarshineeSriram/5)%20Implementing%20MobileNet%20model%20for%20transfer%20learning.ipynb

7) Determining the percentage of NSFW and SFW content in a video:

For this, I take inspiration from the libraries mentioned in the Phabricator task description: https://phabricator.wikimedia.org/T264050. I use the **PySceneDetect library**.

I use the following to extract frame with content-aware detection feature of PySceneDetect (which reduces the number of extracted frames by a huge margin, compared to a generic video frame extractor, and hence helps with faster classification).

```
scenedetect --input video1.mp4 detect-content list-scenes save-images
```

After this, I loop through the extracted frames to determine the NSFW and SFW percentages.

Link: https://github.com/HarshineeSriram/Wikimedia-NSFW-Classifier-Reports/blob/main/reports/HarshineeSriram/6)%20NSFW%20content%20detection%20in%20videos.ipynb