RESEARCH PROJECT

WHICH ONE IS BEST FOR CLASSIFICATION: CARTOON OR SKETCH

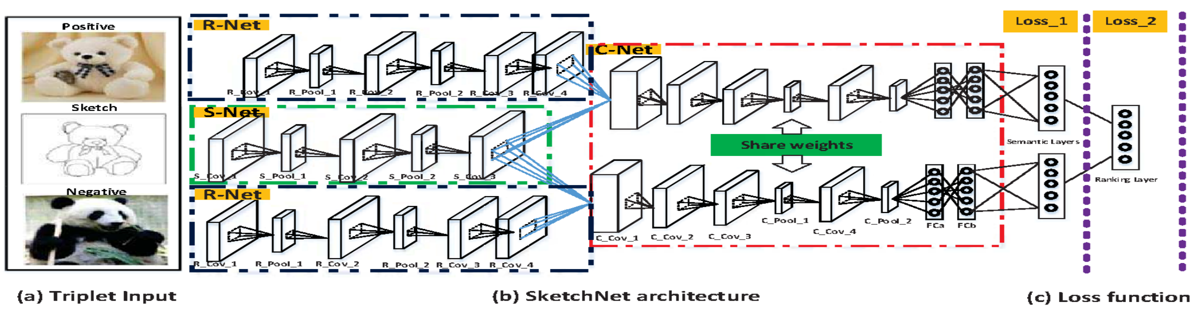
NAME - SAYAK CHAKRABORTY

Name - Sayak Chakraborty

REVIEW

1)CARTOON FACE IMAGE CLASSIFICATION: The proposed system focuses on cartoon face recognition and understanding thereby categorizing the difference between cartoon and photos and performing classification tasks. In this task, a database containing cartoon images with many intra class variations is used thereby making the dataset challenging for classification tasks. It uses Gabor filters for feature extraction and morphological operations for reducing noise thereby improving the accuracy in classification [1]. It also integrates deep neural networks(ANN) for understanding cartoon emotions on large datasets and enhances the algorithms capability for better analysis of cartoon images[1]. Although the paper presents results in a comprehensive manner, it lacks a detailed discussion of the limitations and challenges of the algorithm and also evaluation of performance metrics like accuracy, precision, etc[3]. Also, the paper does not include comparison between the existing methods as this would help in demonstrating the novelty and superiority of the proposed system when compared to the recent state of art approaches[2]. The given work has many applications in entertainment, education etc.

2)SKETCHNET-SKETCH CLASSIFICATION WITH WEB IMAGES:In this paper, a weakly supervised approach called sketchnet is used for sketch classification using web images[4]. Although both the papers aim to improve image classification, the latter performs better in terms of performance metrics. The cartoon face image classification paper uses ANN[1] neural network for classification and concluded that ANN based classification[1] is effective for cartoon face image retrieval whereas the sketchnet paper proposed a CNN trained on triplets of sketch, positive and negative image data[4]. This approach helped the network to learn latent structures between sketch and real images thereby resulting in better accuracy[4]. However, there are still limitations in the classification performance in certain class categories.



CITATIONS

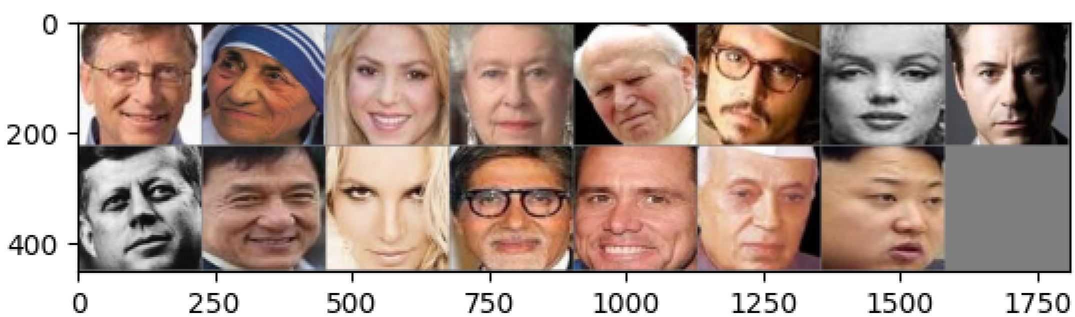
1)Chaitrashree,Aishwarya Ha,Aishwarya K,Deepa,Surya et al. Cartoon Face Image Classification. IRJMETS 05, 2582-5208(2023)
2)Jain, N., Gupta, V., Shubham, S. et al. Understanding cartoon emotion using integrated deep neural network on large dataset. Neural Comput & Applic 34, 21481–21501 (2022).
3)Lee, HS., Kang, BY. Continuous emotion estimation of facial expressions on JAFFE and CK+ datasets for human–robot interaction. Intel Serv Robotics 13, 15–27 (2020).
4) H. Zhang, et al., "SketchNet: Sketch Classification with Web Images," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016 pp. 1105-1113.doi: 10.1109/CVPR.2016.125

INTRODUCTION

Cartoon and Sketches classification with respect to real images can be a very challenging task as it often lacks realistic details like details, texture etc. and has large exaggerated features.
In this paper, we have used the same IIIT-CFW dataset but here we mainly dealt with the real faces dataset of the celebrities and not with the already defined caricature dataset. Why so?

1. The challenging part of the IIT-CFW caricature dataset is that the figures are unaligned and unannotated and not of proper size leading to loss of spatial Information thereby hindering the model's ability to generalize unseen examples.
2. Preprocessing to align and annotate images can be a complex task and requires higher GPU usage which may be expensive.
3. Training a model on diverse representations of the same content can enable the use of transfer learning. Features learned from one style(cartoons) may transfer well to the other style(sketches) leading to improved performance.

Thus in this project, we have used the real face dataset of the IIIT-CFW dataset and performed translation on these images to convert them into cartoons and sketches. We then used a CNN based model which is VGG-16 in this project, for classification of these images. The overall accuracy that we got in classifying sketches with respect to real images outperformed cartoon images but still the results are not as good as expected.



IIIT-CFW REAL IMAGE DATASET(BATCH SIZE = 15)

METHODOLOGY

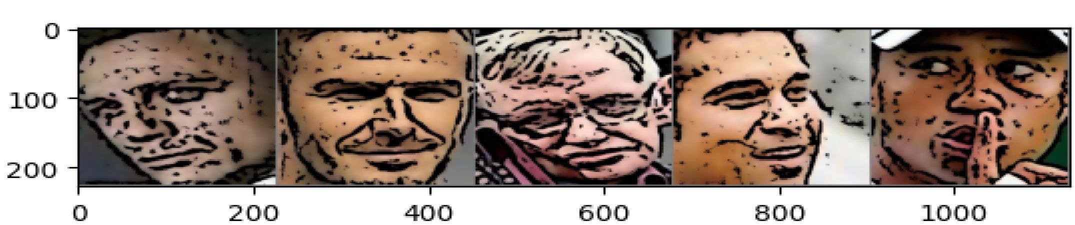
1.**Create the Dataset**:

The IIIT-CFW dataset consists of 1000 real images containing 10 images of each celebrity. Therefore we can say that there are 100 classes in this dataset. Create subfolders of each class and then perform image translation on these images to cartoon and sketches and store them in their respective classes. This will help in organizing a large dataset into a structured hierarchy thereby making it easier to process and manage data of the classes. Also it helps in facilitating model training and enhancing readability and interpretability of the model.

CREATING CARTOON DATASET FROM THE REAL IMAGE DATASET

```
from PIL import Image, ImageFilter
import cv2

filtered_dataset = '/content/drive/MyDrive/Project/Filtered_Dataset/'
cartoon_path = '/content/drive/MyDrive/Project/Cartoon/'
cartoon_classfolder = '/content/drive/MyDrive/Project/Cartoon_Class/'
file =[f for f in os.listdir(filtered_dataset)]
for imgs in file:
 source_path = os.path.join(filtered_dataset,imgs)
 image = cv2.imread(source_path)
 smooth = cv2.bilateralFilter(image,d=9,sigmaColor=75,sigmaSpace=75)
 gray = cv2.cvtColor(smooth,cv2.COLOR_BGR2GRAY)
 blurred = cv2.medianBlur(gray,7)
edgemask=cv2.adaptiveThreshold(blurred,255,cv2.ADAPTIVE_THRESH_MEAN_C,cv2.
THRESH_BINARY,9,2)
 edge_image = cv2.cvtColor(edgemask,cv2.COLOR_GRAY2BGR)
 cartoon = cv2.bitwise_and(smooth,edge_image)
 output_path = os.path.join(cartoon_path,imgs)
 cv2.imwrite(output_path,cartoon)
cartoon_classfolder = '/content/drive/MyDrive/Project/Cartoon_Class/'
for name in classes:
 class_folder = os.path.join(cartoon_classfolder,name)
 os.makedirs(class_folder, exist_ok=True)

file =[f for f in os.listdir(cartoon_path)]
folder = [f1 for f1 in os.listdir(cartoon_classfolder)]

for files in file:
 for folders in folder:
   if folders == files[:-8]:
     src_path = os.path.join(cartoon_path, files)
     dst_folder = os.path.join(cartoon_classfolder,folders)
     shutil.copy(src_path,dst_folder)
```

The process involves reduction in texture to eliminate fine details. Here we have used bilateral filtering as it considers both spatial and intensity while preserving edges. We have enhanced the contrast of different regions to make the cartoons more pronounced. Used adaptive threshold to segment images based on the region of interest and identified prominent edges and performed image masking using bitwise and operator. The results are shown below.



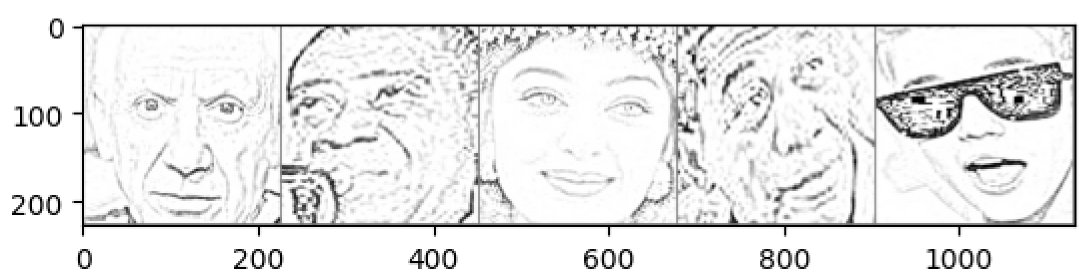

```
Cartoon Dataset
DanielCraig DavidBeckham StephenHawking SachinTendulkar TigerWoods
```

The created cartoon dataset(batch size=5)

CREATING SKETCH DATASET FROM THE REAL IMAGE DATASET

```
from PIL import Image, ImageFilter
import cv2

filtered_dataset = '/content/drive/MyDrive/Project/Filtered_Dataset/'
sketch_path = '/content/drive/MyDrive/Project/Sketch/'
sketch_classfolder = '/content/drive/MyDrive/Project/Sketch_Class/'

file =[f for f in os.listdir(filtered_dataset)]
for imgs in file:
 source_path = os.path.join(filtered_dataset,imgs)
 image = cv2.imread(source_path)
 gray = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
 gray_inverse = 255 - gray
 blur = cv2.GaussianBlur(gray_inverse,(21,21),0)
 blur_inverse = 255 - blur
 sketch = cv2.divide(gray, blur_inverse, scale = 256.0)
 output_path = os.path.join(sketch_path,imgs)
 cv2.imwrite(output_path,sketch)

for name in classes:
 class_folder = os.path.join(sketch_classfolder,name)
 os.makedirs(class_folder, exist_ok=True)

file =[f for f in os.listdir(sketch_path)]
folder = [f1 for f1 in os.listdir(sketch_classfolder)]

for files in file:
 for folders in folder:
   if folders == files[:-8]:
     src_path = os.path.join(sketch_path, files)
     dst_folder = os.path.join(sketch_classfolder,folders)
     shutil.copy(src_path,dst_folder)
```

Here we convert the grayscale images taken as input and inverse them and use Gasussian blur on them. The purpose of this is to reduce noise, enhance edges thereby leading to smoother and appealing sketches.




```
Sketch Dataset
PabloPicasso LouisArmstrong AishwaryaRai MorganFreeman JustinBieber
```

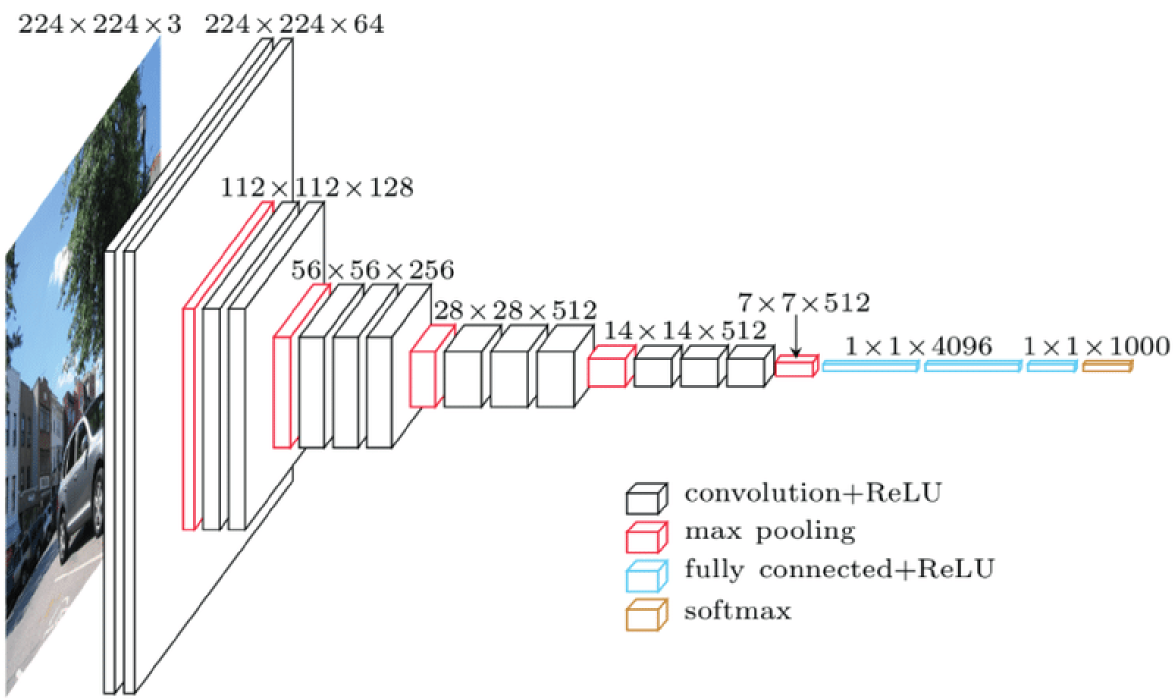
The created sketch dataset(batch size=5)

2.**VGG-16 model:**

The VGG-16 model is a 16 layer convolutional neural network model used in the classification of images. It consists of 3*3 small convolutional layers followed by 2*2 max pooling layers thereby making the model better in terms of classification. The accepted input size of the image is 224*224 for this model.Since the images in the dataset are not of proper size, we had to reshape them in order to build our new dataset. We are using this model in this work for the classification of cartoon images and sketches with respect to the real images.

Advantages of the model:
It is often used in the field of transfer learning as it helps in enhancing performance of the model. The stacking of multiple convolutional layers helps in learning complex patterns thereby making it effective for image classification tasks.

Disadvantages of the model:
Due to model depth and high number of parameters, it is computational inefficient. Since the size of our dataset is small, this model can cause high chances of overfitting thereby hampering the accuracy of the model.



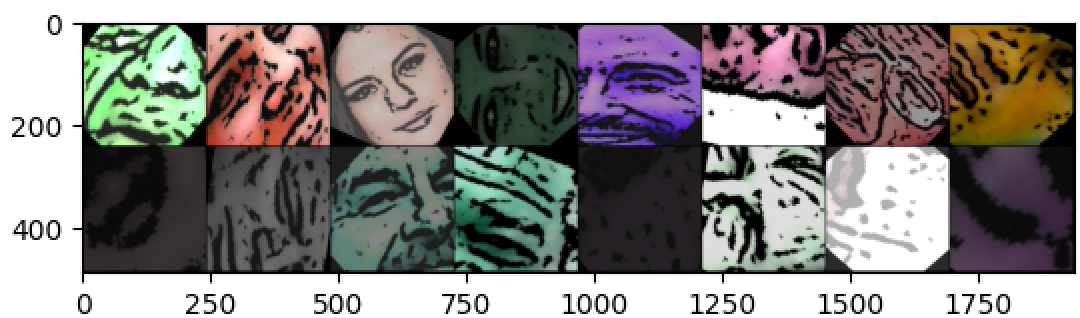
The model

3.**Increase the size of the dataset to avoid overfitting:**

As the size of the dataset is not large, there is a high chance for the model to overfit leading to incorrect accuracies. To get rid of these, we have used image augmentation like flip,crop,blur,feature extraction etc to increase the size of the dataset and introduced early stopping in the validation set to break the training of the dataset.

```
transform =
transforms.Compose([transforms.RandomRotation(50),transforms.RandomResized
Crop(240),transforms.RandomHorizontalFlip(),transforms.ColorJitter(brightn
ess=0.8, contrast=0.8, saturation=0.8, hue =0.8),transforms.ToTensor(),])
```

A series of image transformations is defined in the pytorch's transforms.Compose where Transformations include random rotations,cropping, horizontal or vertical flipping and color jitter is used to contrast enhancing, saturate and hue the images and then convert them into a tensor which can be fit into a CNN model for classification. Here we ran a loop from 12 to 51(40 images) on each image of each class so that the number of images in each class is equal. This is also done to get better accuracy and avoid overfitting. The results shown below are images of a few random classes of the transformed dataset.



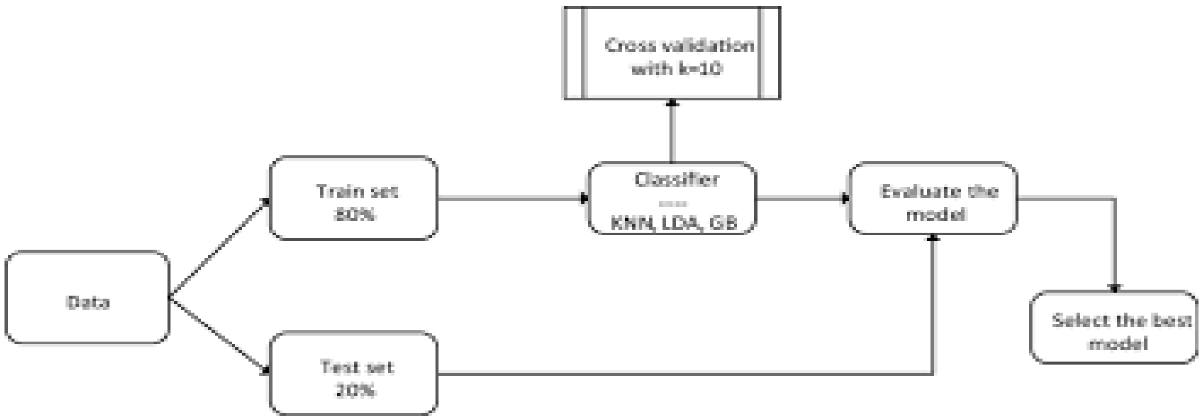
The created transformed dataset(batch size=16)

Ground Truth:`NelsonMandela KateMiddleton SelenaGomez PrincessDiana HughJackman SylvesterStallone NelsonMandela TigerWoods PrinceCharles JohnFKennedy PeterJackson MorganFreeman AbrahamLincoln TomCruise LanceArmstrong MichaelJordan`

4. **Train/Test the model for classification:**

Now that our dataset is ready, we can train our model with respect to the dataset that we have created. We have trained the model for both cartoons and sketches. The GPU usage was relatively high for both cartoons and sketches. Training time for cartoons was approximately 1 hr whereas for sketches it was around 35 minutes keeping all the hyper parameters fixed. This could be because of intricate details and patterns in the cartoon dataset as there could be any noisy, robust images. Also sketches appear to converge faster than cartoons.

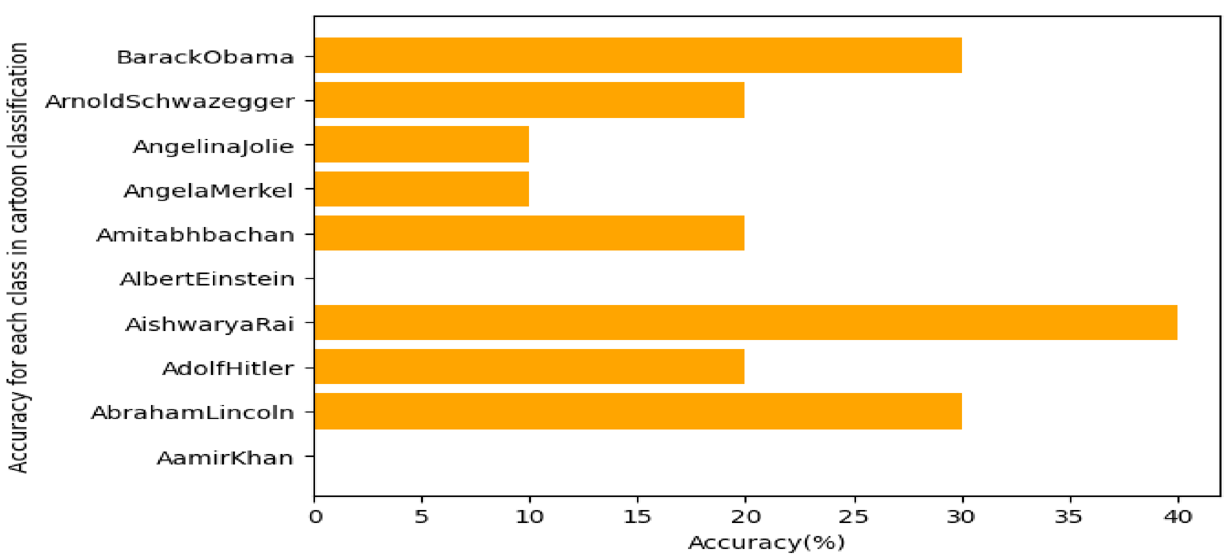
To avoid overfitting, we have created a validation set which will break the iteration whenever the model is found to converge.



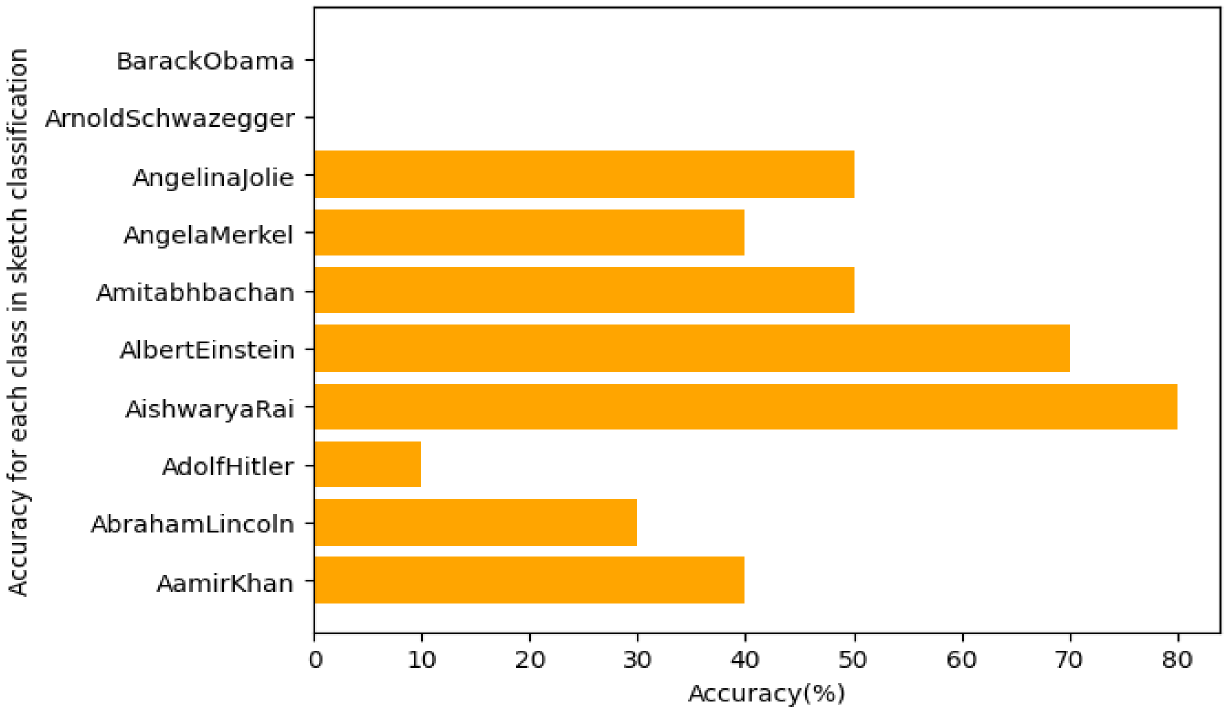
RESULTS

We performed classification operations for both cartoon images and sketches in VGG-16 and got an overall accuracy of 25% in cartoons and 38% in sketches with respect to the original real image dataset.Examples of the accuracies of the first 10 classes are provided.

For Cartoons:



For Sketches:



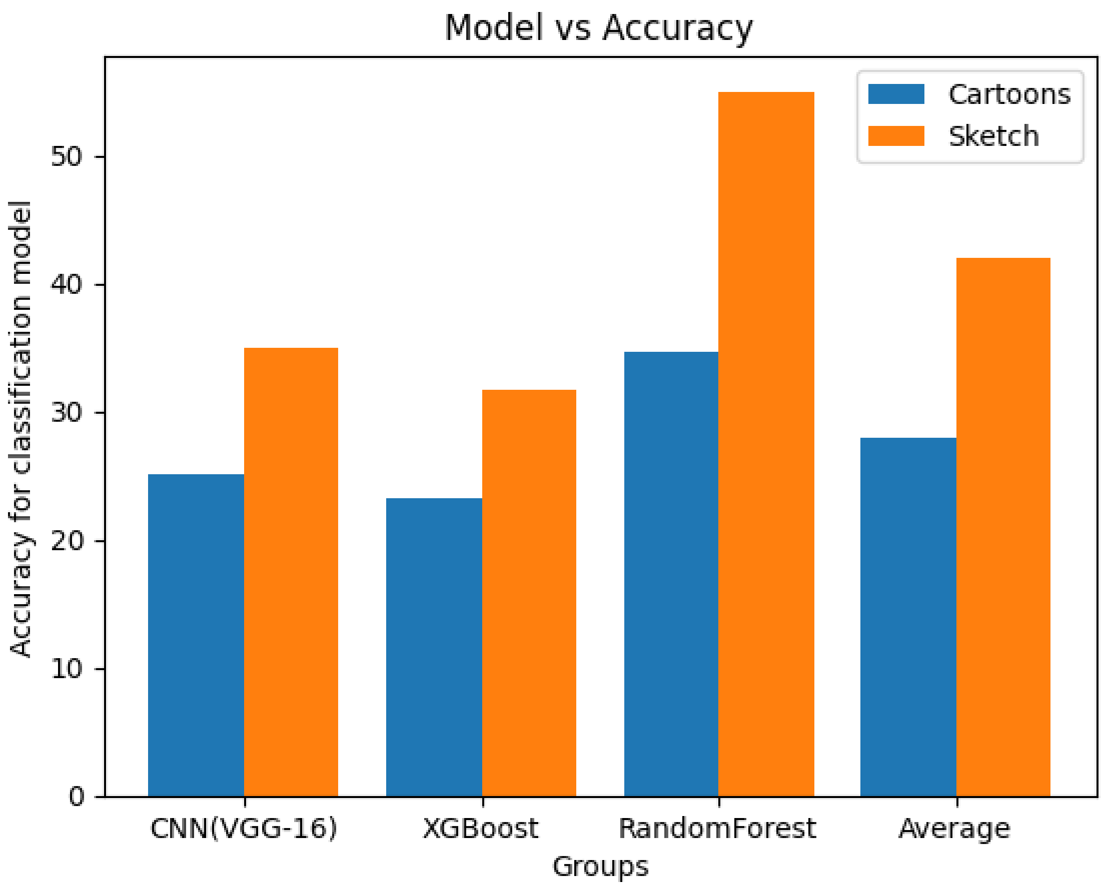
Hence from our results we can say that sketches outperformed cartoon images while classifying against real images.

Why so?

1. Cartoons might be highly abstract with exaggerated features whereas sketches may have different levels of details ranging from refined to rough features.
2. Cartoons can have non-realistic color plates while sketches have grayscale plates. Since real images include a wide range of natural colors, it makes the classifier quite discern as it has to learn a lot of unique features.
3. Cartoons simplify complex objects whereas sketches contain more details of the object as it preserves edges and features of the original image.
4. The imbalance in the cartoon and sketch dataset can also be one of the reasons for low accuracy.
5. This could also be one of the reasons that the total number of images in the dataset is not enough for classification models like VGG-16, CNN as small dataset can lead to high overfitting.Sometime the model cannot figure out essential features present in the targeted population.

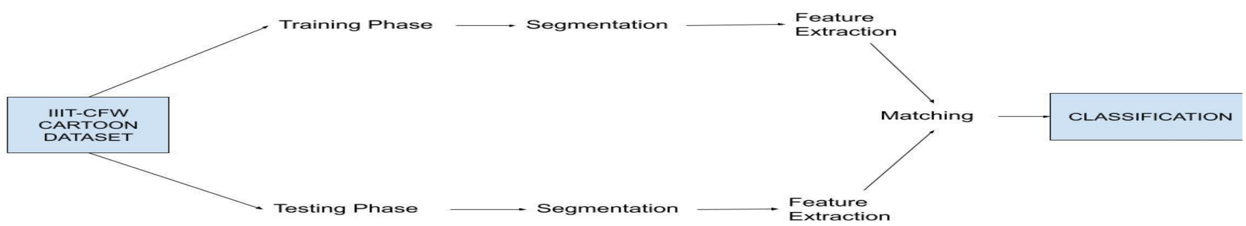
Lets try this classification with some machine learning models(since they perform well on small dataset and due to our limited GPU resource) to figure out if classification performs better than deep learning models.

MODEL	CARTOON	SKETCH
VGG-16	25%	38%
XGBoost	23.3%	31.7%
RandomForest	34.6%	55%
Average Performance(Approx)	28%	42%



DISCUSSION

Since cartoon images are distorted and exaggerated, we can use the proposed system as mentioned in the paper “CARTOON FACE IMAGE CLASSIFICATION”. Using segmentation and applying Gabor filters, we can extract meaningful features and then apply morphological operations to reduce noise which when trained afterwards can lead to better accuracy during classifications.



Comparing sketches, we can use the methodology as discussed in the paper "SKETCHNET-SKETCH CLASSIFICATION WITH WEB IMAGES" where they trained the model on sketches, positive and negative images i.e. on all three layers which helped the model to learn latent structures and features thereby resulting into better accuracies.

Other ways would be to use Image translation techniques, like Generative Adversarial Network, Pix2Pix, Neural Style Transfer etc. This model can help in improving the accuracy but they are quite complex. Training such models is way more complex and time consuming if high end GPUs are not used. Training such models from scratch is not feasible as it requires high computational resources and clusters.

Creating a larger dataset using data augmentation, using large CNN complex models like Resnet, vgg-50 and fine tuning hyperparameters could be some other ways of improving the accuracy.

CONCLUSION

Even though we had limited GPU resources, we could train/test our dataset well on small machine learning ensemble models like Random Forest and XGBoost due to the size in dataset. Like the results from the research papers, we can also conclude that Sketches work better than cartoons in terms of classification. Although there are ways to improve the accuracy of these classification models, there will be some restrictions as well like limitations of GPU clusters and other computational resources. The overall motive of this project is to figure out which one classifies better wrt real images as there are wide range of applications:

a)This could be valuable for artistic style analysis and recommendation.

b)Evaluating the performance of cartoons and sketches can serve as a benchmark for assessing the limitations of recent image processing techniques. This can help in better research and development.

c)This may be useful for research involving art and technology and for cultural studies.

REFERENCES

1)O'Shea K, Nash R. An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458. 2015 Nov 26.
2)Chaitrashree,Aishwarya Ha,Aishwarya K,Deepa,Surya et al. Cartoon Face Image Classification. IRJMETS 05, 2582-5208(2023)
3)Lee, HS., Kang, BY. Continuous emotion estimation of facial expressions on JAFFE and CK+ datasets for human–robot interaction. Intel Serv Robotics 13, 15–27 (2020).
4)Jain, N., Gupta, V., Shubham, S. et al. Understanding cartoon emotion using integrated deep neural network on large dataset. Neural Comput & Applic 34, 21481–21501 (2022).
5) H. Zhang, et al., "SketchNet: Sketch Classification with Web Images," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016 pp. 1105-1113.doi: 10.1109/CVPR.2016.125
6)D. I. Swasono, H. Tjandrasa and C. Fathicah, "Classification of Tobacco Leaf Pests Using VGG16 Transfer Learning," *2019 12th International Conference on Information & Communication Technology and System (ICTS)*, Surabaya, Indonesia, 2019, pp. 176-181, doi: 10.1109/ICTS.2019.8850946.
7)https://scikit-learn.org/stable/modules/neural_networks_supervised.html#classification
8)https://scikit-learn.org/stable/modules/ensemble.html#random-forests-and-other-randomized-tree-ensembles
9)https://scikit-learn.org/stable/modules/sgd.html#classification
10)https://scikit-learn.org/stable/modules/multiclass.html#multiclass-classification