Visual Recognition Project on Image Classification Between Selfie and Non-Selfie Image.

Project supervisor-Dr. Shiv Ram Dubey

1st Sameer Ahmed	2nd Rahul	3rd Sanjeet Beniwal	4th Jitu Rajak	5th Harsh garg
B.tech IT	*B.tech IT*	*B.tech IT*	*B.tech IT*	*B.tech IT*
IIIT Alllahabad	*IIIT Alllahabad*	*IIIT Alllahabad*	*IIIT Alllahabad*	*IIIT Alllahabad*
UP, INDIA	UP, INDIA	UP, INDIA	UP, INDIA	UP, INDIA
iit2020053@iiita.ac.in	iit2020244@iiita.ac.in	iit2020052@iiita.ac.in	iit2020218@iiita.ac.in	iit2020082@iiita.ac.in

***Abstract*—In the era of ubiquitous smartphone usage and social media platforms, the line between personal and non-personal images has blurred significantly. Selfies, which are self-portraits typically taken with a smartphone camera, have become a ubiquitous form of self-expression. However, distinguishing between selfie and non-selfie images automatically presents an interesting and challenging problem in computer vision. This semester project targets to address the problem by developing a robust image classification system that can accurately differentiate between selfie and non-selfie images.**

***Index Terms*—Vgg-16, VGG-19, ResNet-50, Visual Recognition, Binary Classification, Supervised Learning**

I. Introduction

the lines between personal and impersonal images have blurred. The proliferation of selfies—mostly self-portraits taken with a smartphone camera—has redefined the way we express ourselves. Therefore, the need to distinguish selfies from non-selfie images has become an interesting and challenging problem in computer vision. This project aims to solve this problem by creating a robust image classification system that can distinguish two different groups of images.

The motivation behind this effort stems from the increase in visual content sharing on social media platforms and the desire to speed up the sharing process of image number 2. Selfies are often filled with personal and emotional meaning and are often shared on platforms such as Instagram, Facebook and Snapchat. The ability to distinguish selfies from other types of photos could have many uses, from content recommendations to advertising campaigns and even security and privacy concerns.

A good method will be used to solve this problem. The first step involves collecting and processing different files containing different types of selfie and non-selfie images. This information will form the basis for training and analysis of computer models.
In the development of computer vision models, self-tracking techniques such as random learning or tolerance can be chosen. Self-monitoring aims to enable the model to learn the content of image data without relying on external text. At this stage, the model will focus on understanding the features of the image that are important in understanding the features that distinguish the individual from other images.
The model will go through the improvement process. Head classification will be built into the model and the neural network will be trained on recording data that includes selfie and non-selfie photos. This stage allows the model to be diversified according to the examined features.
The final evaluation of the model's performance will be made on a separate data set using the model distribution. This rigorous evaluation will provide insight into the field of model development or preprocessing to improvise the accuracy and robust behaviour of the model in distinguishing individuals from images that are not a selfie.
In summary, computer vision models that produce accurate and robust models for distinguishing selfies from non-selfie images show broad promise in social media, user experience, and data analysis. The program aims to solve problems arising from this exciting and changing problem and contribute to the advancement of computer vision in today's digital world.

Identify applicable funding agency here. If none, delete this.

II. Problem Statement

Develop an accurate and robust computer vision model capable of distinguishing between selfie and non-selfie images.

A. Approach

To address the challenge of classifying selfie and non-selfie images, following steps will be employed. First, a dataset containing a wide range of selfie and non-selfie images will be collected and preprocessed. This dataset will serve as the foundation for training and evaluation. Next, a self-supervised learning technique, such as contrastive learning or pretext tasks, will be chosen to train a neural network model. This self-supervised task will encourage the model to learn meaningful representations from the image data without relying

on external labels. During this phase, the model will focus on understanding the intrinsic features of the images. After successful self-supervised learning, the model will undergo fine-tuning. A classification head will be added to the model, and the network will be trained on the labeled data, which includes both selfie and non-selfie images. Once the model is trained, it will be evaluated using standard classification metrics on a separate test dataset. Performance analysis will help identify areas for improvement to the model or data preprocessing techniques.

III. Literature Review

A. Paper 1

- **Author–** Shiyun Kong
- **Title–** "Self-supervised Image Classification Using Convolutional Neural Network"
- **Method/ approach used–** "Self-supervised framework SimCLR on image classification successfully clusters a large number of images into an optimum amount categories"
- **Achieved Performance –** "The accuracy on classifying Mnist dataset is 32
- **Advantages–** The model is based on the Simple Framework for Contrastive Learning of Visual Representations (SimCLR), The Base Model used in this experiment is the ResNet, the PCA algorithm is used to reduce dimension whilst preserving the feature of original data, then K-means is used for clustering data into groups
- **Dataset–** Mnist Dataset
- **Scope–** The accuracy of the image classification with semantically similar pictures should improve. SimCLR model is formed based on the assumption that there is a balanced distribution of different kinds of images across the entire dataset, unlike the everyday situation, where people cannot guarantee the distribution.
- **Year–** 2023

B. Paper 2

- **Author–** Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, Ross Girshick
- **Title–** "Momentum Contrast for Unsupervised Visual Representation Learning"
- **Method/ approach used–** The key method employed in this paper is Momentum Contrast (MoCo), which is introduced as an unsupervised learning framework for training visual representations. MoCo leverages a momentum encoder to create a dynamic dictionary of negative samples for each image in the dataset. It utilizes a queue mechanism to store a history of negative samples, improving the quality of contrastive learning. Additionally, MoCo introduces a novel use of the stop-gradient operation for contrastive learning.
- **Achieved Performance –** The paper demonstrates state-of-the-art results on multiple benchmark datasets, including ImageNet, where it surpasses previous unsupervised learning approaches in terms of representation learning performance. It shows that the learned representations generalize well to downstream tasks like object detection and semantic segmentation.
- **Advantages–** - Efficiency: Momentum Contrast is computationally efficient, making it a practical choice for large-scale unsupervised learning tasks. - Improved Representations: The approach improves upon the quality of learned representations, leading to better generalization in various downstream tasks. - Reduced Labeling Costs: By enabling effective unsupervised learning, the method reduces the reliance on costly labeled data, making it particularly valuable in scenarios where labeled data is scarce.
- **Dataset–** The primary dataset used for evaluation in the paper is ImageNet, a large-scale image classification dataset. The effectiveness of the Momentum Contrast approach is demonstrated on this dataset, among others.
- **Scope–** The scope of this paper is focused on the domain of unsupervised visual representation learning. It addresses the challenge of learning meaningful representations from unlabeled image data, which has broad applications in computer vision and deep learning.
- **Year–** 2020

C. Paper 3

- **Author–** Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey Hinton
- **Title–** "SimCLR: A Simple Framework for Contrastive Learning of Visual Representations"
- **Method/Approach–** SimCLR employs a straightforward and intuitive approach to unsupervised learning. It utilizes a contrastive loss function that encourages the model to bring representations of similar images closer while pushing representations of dissimilar images apart. A crucial aspect of SimCLR is data augmentation, where each image is randomly augmented to create positive and negative pairs. The learned representations are then used for downstream tasks.
- **Achieved Performance–** SimCLR has demonstrated remarkable performance in unsupervised visual representation learning. It has achieved state-of-the-art results on various benchmark datasets, such as CIFAR-10, ImageNet, and more. The learned representations generalize well to a wide range of downstream tasks, including object detection, image classification, and semantic segmentation.
- **Advantages–** - Simplicity: SimCLR's simplicity makes it easy to understand and implement, while still achieving state-of-the-art performance. - Strong Generalization: The learned representations have shown strong generalization capabilities across a variety of downstream tasks. - Lack of Labeled Data Dependency: Like Momentum Contrast, SimCLR reduces the reliance on labeled data, which can be expensive and limited in availability.
- **Dataset–** SimCLR has been evaluated on several datasets, including CIFAR-10, ImageNet, and others. ImageNet, in

particular, serves as a significant benchmark for evaluating the quality of learned representations.
- **Scope–** The scope of this paper is also focused on the field of unsupervised visual representation learning. It addresses the need for robust and generalized feature learning from unlabeled image data, making it highly relevant to various computer vision tasks.
- **Year–** 2020

D. Paper 4

- **Author–** Alexander Kolesnikov*, Xiaohua Zhai*, Lucas Beyer*
- **Title–** "Revisiting Self-Supervised Visual Representation Learning"
- **Method/Approach–** The paper revisits numerous previously proposed self-supervised models and common practices in self-supervised visual representation learning.
- **Achieved Performance–** Using the ImageNet model and the rotation method suggested in the paper, it attains a high accuracy of 55.4.
- **Advantages–** This paper provides multiple insights regarding self-supervised learning, including the observations that (1) lessons from architecture design in the fully supervised setting do not necessarily translate to the self-supervised setting; (2) contrary to previously popular architectures like AlexNet, in residual architectures, the final prelogits layer consistently results in the best performance; (3) the widening factor of CNNs has a drastic effect on the performance of self-supervised techniques; and (4) SGD training of linear logistic regression may require a very long time to converge.
- **Scope–** The paper demonstrated that the performance of existing self-supervision techniques can be consistently boosted, leading to a significant reduction in the gap between self-supervision and fully labeled supervision.
- **Year–** 2019

E. Paper 5

- **Author–** Linus Ericsson, Henry Gouk, Chen Change Loy, and Timothy M. Hospedales
- **Title–** "Self-Supervised Representation Learning: Introduction, Advances and Challenges"
- **Method/Approach–** This paper discusses self-supervised representation learning as an alternative to supervised learning for training deep neural networks. The goal is to overcome the annotation bottleneck by using freely available labels from pretext tasks to train deep representations. This approach has shown promise across various data modalities, including image, video, sound, text, and graphs.
- **Achieved Performance–** This paper highlights the success and potential of Self-Supervised Representation Learning (SSRL) in various applications, including improved performance, generalization, multimodal learning, and data efficiency.
- **Advantages–** - SSRL has made significant contributions to problems involving multiple languages, enabling better performance in tasks like classification and question answering for low-resource languages. - SSRL can mitigate the negative impact of label noise on supervised target tasks, resulting in better performance compared to training on noisy labeled data.
- **Scope–** In this paper, we gain insights into the workings of self-supervised learning and also compare this learning approach to other machine learning algorithms.
- **Year–** 2019

F. Paper 6

- **Author–** Priya Goyal, Dhruv Mahajan, Abhinav Gupta, Ishan Misra
- **Title–** "Scaling and Benchmarking Self-Supervised Visual Representation Learning"
- **Method/Approach–** The paper utilizes two key image-based self-supervised methods: Jigsaw and Colorization. The Jigsaw method involves dividing an image into non-overlapping patches to form a 'puzzle' and learning image representation by solving this puzzle. Colorization involves manipulating the image's color information.
- **Achieved Performance–** The paper reveals that when self-supervised learning approaches, specifically using the Jigsaw and Colorization methods, are scaled to a large number of images (up to 100 million), they can match or even exceed the performance of supervised pre-training. This has potential implications for applications of machine intelligence that can benefit from unsupervised learning methods.
- **Advantages–** - Flexibility to alter problem complexity, making it easier to adjust the learning challenge according to specific needs. - The development of an extensive benchmark model offers a clearer and standardized evaluation mechanism for such learning methods, fostering clarity and progress in the field.
- **Scope–** The self-supervised learning approaches (like Jigsaw and Colorization) can be applied to classify images as selfies or non-selfies based on inherent patterns and features. By scaling up the quantity of training data, we can improve the performance of our detection model.
- **Year–** 2019

IV. Dataset

The dataset for the problem of detecting whether an image is a selfie or not has been sourced from Kaggle. The dataset has been randomly sampled from a bigger dataset containing roughly 70,000 images.

The dataset is categorized into two classes: "Selfie" images and "non-selfie" images. The dataset is further split the data into training set, validation set, and testing set. The training set comprises a total of 10,000 images, while the testing and validation data each consist of 3,930 images.

Dataset link: Kaggle-dataset- link

Samples from the dataset are:
1. Selfie images:


2. Non-Selfie images:


V. Methods

A. CNN Models

VGG-16 model architecture – Karen and Andrew [8] created a 16-layer network comprised of 13 convolutional layers and the number of Fully connected layers is three followed by SoftMax classifier.

The network takes an input image of size 224x224 pixels. The 13 convolutional layers are each followed by a ReLU function, which introduces non-linearity into the model. After some of the convolutional layers, spatial dimensions of representation are reduced using max pooling layers which helps the network focus on the most important features. After the convolutional layers there are 3 fully connected layers that take the extracted features from the convolutional layers and use them for classification. To get class scores we used Softmax function.

Vgg-19 Model Architecture - VGG19 is deeper than VGG16, which helps the model to learn more complex features that VGG16 could not learn. The number of convolutional layers is 16 which are followed by 3 fully connected layers and at last the softmax layer gives the class scores[8].

Deep residual network or ResNet is a model developed by He et al. in 2016 [7] . Deep learning training, in general, time consuming and has a limit of working effectively only up to a certain number of layers, ResNet helped overcame some of these drawbacks by introducing skip connections. The advantage of the ResNet model against other architectural models is the performance of this model in deeper CNN models.In ResNet, Skipping connections are made on two to three layers with ReLU as activation function and batch normalization is also performed[7].

Because the model has been trained on a large dataset, it has learned a good representation of low level features such as spatial, edges, rotation, lighting, and shapes, which can be shared to enable knowledge transfer and act as a feature extractor for new images in various computer vision problems. Despite the fact that these additional images are from completely different categories than our target dataset, the pretrained model should be able to extract useful characteristics from them using transfer learning techniques. In this paper we will unleash the power of transfer learning by using pretrained model - VGG-16, VGG-19 and ResNet 50 models.

VI. Implementation Details

Data Preparation and Data augmentation - Keras uses ImageDataGenerator() to quickly set-up python generators that turn image files into preprocessed tensors that can be fed directly into models during training. Image transformation procedures like as rotation, translation, and zooming are performed on the training dataset to generate new versions of existing images. During training, we injected these fresh photos into our model.

Hyperparameters - We are initializing the model with the weights pre-trained on ImageNet dataset. The input images of the model are of dimensions 224X224 pixels with three colour channels (RGB). In case of model the average pooling layer in the head of model has dimensions 7X7, the first dense layer uses 256 units and uses ReLu activation function. The dropout rate is 0.5. The final dense layer has 2 units and activation function is Softmax. The optimizer used here is Adam optimizer with learning rate as 1e-4. The loss function here is binary cross entropy loss, and the Batch Size is 32.

VII. Results

	precision	recall	f1-score	support
NonSelfie	0.99	0.98	0.98	1966
Selfie	0.98	0.99	0.98	1966
accuracy			0.98	3932
macro avg	0.98	0.98	0.98	3932
weighted avg	0.98	0.98	0.98	3932

Fig. 1. Results of VGG-16

Fig. 2. Loss and Accuracy for VGG 16

	precision	recall	f1-score	support
NonSelfie	0.99	0.98	0.99	1966
Selfie	0.98	0.99	0.99	1966
accuracy			0.99	3932
macro avg	0.99	0.99	0.99	3932
weighted avg	0.99	0.99	0.99	3932

Fig. 3. Results of VGG 19

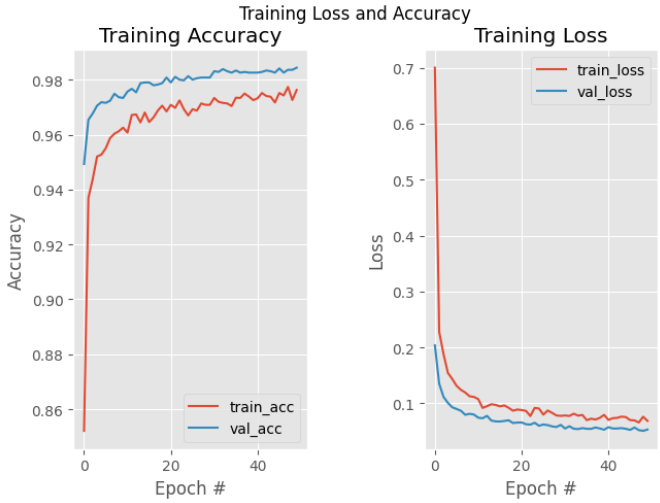


Fig. 4. Graph of VGG 19

	precision	recall	f1-score	support
NonSelfie	0.99	0.99	0.99	1966
Selfie	0.99	0.99	0.99	1966
accuracy			0.99	3932
macro avg	0.99	0.99	0.99	3932
weighted avg	0.99	0.99	0.99	3932

Fig. 5. Results of ResNet-50

Fig. 6. Graph Of ResNet-50

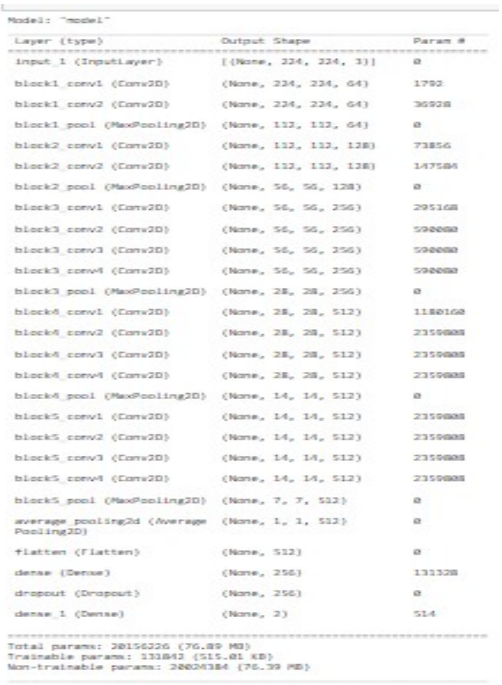
VIII. Appendix 1 - Summary of Models Used

A. VGG16

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 vgg16 (Functional)          (None, 7, 7, 512)         14714688

 flatten (Flatten)           (None, 25088)             0

 dropout (Dropout)           (None, 25088)             0

 dense (Dense)               (None, 1)                 25089

=================================================================
Total params: 14,739,777
Trainable params: 25,089
Non-trainable params: 14,714,688
_________________________________________________________________
```

Fig. 7. VGG16 Model Architecture

B. VGG19



```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]     0

 block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792

 block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928

 block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0

 block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856

 block2_conv2 (Conv2D)       (None, 112, 112, 128)     147584

 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0

 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168

 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv4 (Conv2D)       (None, 56, 56, 256)       590080

 block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0

 block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160

 block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv4 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0

 block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv4 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0

 average_pooling2d (Average  (None, 1, 1, 512)         0
 Pooling2D)

 flatten (Flatten)           (None, 512)               0

 dense (Dense)               (None, 256)               131328

 dropout (Dropout)           (None, 256)               0

 dense_1 (Dense)             (None, 2)                 514

=================================================================
Total params: 20156226 (76.89 MB)
Trainable params: 131842 (515.01 KB)
Non-trainable params: 20024384 (76.39 MB)
```

Fig. 8. VGG19 Model Architecture

IX. Appendix 2 - Models Performing on Real Dataset

A. VGG16

Fig. 9. Performance of VGG16 on Real Dataset

B. VGG19

Fig. 10. Performance of VGG19 on Real Dataset

C. Resnet

Fig. 11. Performance of VGG19 on Real Dataset

X. References

1) S. Kong, "Self-supervised Image Classification Using Convolutional Neural Network," 2023 IEEE 3rd International Conference on Power, Electronics and Computer Applications (ICPECA), Shenyang, China, 2023, pp. 1283-1290, doi: 10.1109/ICPECA56706.2023.10075949.
2) He, K., Fan, H., Wu, Y., Xie, S., Girshick, R. (2020). Momentum contrast for unsupervised visual representa-

tion learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 9729-9738).
3) Chen, T., Kornblith, S., Norouzi, M., Hinton, G. (2020, November). A simple framework for contrastive learning of visual representations. In International conference on machine learning (pp. 1597-1607). PMLR.
4) Kolesnikov, A., Zhai, X., Beyer, L. (2019). Revisiting self-supervised visual representation learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 1920-1929).
5) Ericsson, L., Gouk, H., Loy, C. C., Hospedales, T. M. (2022). Self-supervised representation learning: Introduction, advances, and challenges. IEEE Signal Processing Magazine, 39(3), 42-62.
6) Goyal, P., Mahajan, D., Gupta, A., Misra, I. (2019). Scaling and benchmarking self-supervised visual representation learning. In Proceedings of the ieee/cvf International Conference on computer vision (pp. 6391-6400).
7) He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
8) Simonyan, Karen, and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition." ArXiv, (2014). Accessed November 2, 2023. /abs/1409.1556.
9) Dataset - https://www.kaggle.com/datasets/jigrubhatt/selfieimagedetectiondataset