Human Action Detection: ResNet50 CNN MODEL

Sejal Patil[1]

Other group members:
Sneha Naidu,[2] **Aman Afthab**,[3] **Louise Williams**[4]

Abstract. In this study, we have used Datasets to evaluate the performance of ResNet model. One of the current study fields in computer vision is human activity recognition, which has applications in security monitoring, healthcare, and human computer interface. It can be difficult to identify human activity from video or still images because of issues such backdrop clutter, partial occlusion, changes in scale, perspective, lighting, and look. For determining a person's physical activity, neural networks are the ideal algorithms. In this paper, we understand the application of a convolutional neural network (CNN), more precisely the ResNet50 model, trained for image classification, to the detection of human activity. With the use of our own dataset, we trained the model in this study, allowing us to compare the model's accuracy and resilience against other models developed by my team members. The ResNet50 model, with an accuracy of 26.77

1 Introduction

Human activity detection has a wide range of uses because of its impact on wellbeing. It is becoming a fundamental tool in healthcare solutions, such as preventing obesity and caring for the elderly. Activity detection is the basis for the development of many potential applications in health, wellness, and sports: monitor health, discover activity patterns and improve wellbeing. Neural networks are the perfect algorithms to determine a person's physical activity. This is due to their ability to recognize the patterns behind the data.

Human action detection is a subfield of computer vision that analyses photos to determine what humans are doing. The single-image action recognition has broadly classified into three categories: the method based on the entire image, the method based on human position, and the method based on human-object interaction. The method based on the entire picture is to take the entire image as a classification issue, extract the features via the neural network layer, and then use the classifier to determine the label category.

There are two key queries across different categorization techniques: "What action?" (specifically, the issue with recognising) and "Where in the image?" (Specifically, the localization issue). The kinetic states of a person must be known when trying to detect human actions so that the computer can do so effectively. Human actions like "walking" and "running" come up pretty regularly in daily life and are rather simple to identify. The identification of more intricate tasks, such as "peeling an apple," is more challenging. Simpler, more easily recognisable activities can be created by breaking down more complex ones. Usually, identifying things in a scenario may help us better comprehend human behaviour since it can provide us important details about what is happening right now.

When the image was created, we added the expected result category to the picture after marking the humans in the image using the human body detection bounding box in the dataset.Different body motions, contrasting backdrop lighting, and blocked and unfinished characters can all be seen in the image.A human body that is partial or obstructed, for instance, depicts the predicament of several people.In these images, the network has produced positive results for recognition.[6] In this study, the action recognition challenge is

Figure 1. Representative frames of the main human action classes for various datasets

based on static single pictures. The classification network ResNet-50 is trained. Meanwhile, in this article, the upgraded Mobile Net and VGG16 were introduced to the fused network, which enhanced the performance of action identification and classification accuracy by managing the number of overall parameters. According to the experimental results, the upgraded model's classification indices are all higher than the prior model's. The network has produced good recognition results in the scenario of varied body motions, changing backdrop light, and blocked and partial characters in the images. [1]

[1] School of Computing and Mathematical Sciences, University of Greenwich, London SE10 9LS, UK, email: one@gre.ac.uk
[2] School of Computing and Mathematical Sciences, University of Greenwich, London SE10 9LS, UK, email: two@gre.ac.uk
[3] School of Computing and Mathematical Sciences, University of Greenwich, London SE10 9LS, UK, email: three@gre.ac.uk
[4] School of Computing and Mathematical Sciences, University of Greenwich, London SE10 9LS, UK, email: three@gre.ac.uk

2 Background

ResNet stands for Residual Network and is a specific type of convolutional neural network (CNN) introduced in the 2015 paper "Deep Residual Learning for Image Recognition" by He Kaiming, Zhang Xiangyu, Ren Shaoqing, and Sun Jian. CNNs are commonly used to power computer vision applications. ResNet-50 is a 50-layer deep convolutional neural network. A pretrained version of the network trained on over a million photos from the ImageNet database may be loaded . The pretrained network can classify images into 1000 different object categories, such as keyboards, mice, pencils, and other animals.[4]

The ease with which models may be created using the Keras deep learning API has made it popular. Resnet50 is one of the many pretrained models that are included with Keras and are available for use in experiments by anyone. Consequently, creating a residual network in Keras for computer vision applications like picture classification is not too difficult. Simple instructions are all that are required.[2]

ResNet-34, a version of the original ResNet design, included 34 weighted layers. By utilising the idea of shortcut connections, it offered a creative solution for expanding the number of convolutional layers in a CNN without encountering the vanishing gradient issue. A shortcut link turns a conventional network into a residual network by "skipping over" some levels.

The model was trained using the ImageNet dataset, which comprises a huge number of pictures tagged with their associated classifications. This allowed the model to learn the characteristics and patterns associated with various types of items. The ResNet50 model was then tested on a set of photos to assess its performance on tasks involving image classification. ResNet-50 has fourty-eight convolutional layers, one MaxPool layer, and one average pool layer. Residual neural networks (RNNs) are artificial neural networks (ANNs) that build networks by layering residual blocks. The 50-layer ResNet architecture contains the following features, as illustrated in the table below:

A convolution of 7×7 kernels and 64 additional kernels with a 2-sized stride.A maximum pooling layer with two strides.9 additional layers—3×3,64 kernel convolution, 1×1,64 kernel convolution, and 1×1,256 kernel convolution. These three levels are repeated three times.Iterated 12 additional layers with 1×1,128 kernels, 3×3,128 kernels, and 1×1,512 kernels. Iterated 6 times on 18 additional layers with 1×1,256 cores and 2 cores 3×3,256 and 1×1,1024.9 additional layers with 1×1,512, 3×3,512, and 1×1,2048 cores iterated three times.(The network has 50 levels up to this point).Using the softmax activation function, average pooling is followed by a fully linked layer with 1000 nodes.[5]

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2.x	56×56	[3×3, 64; 3×3, 64] ×2	[3×3, 64; 3×3, 64] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3
conv3.x	28×28	[3×3, 128; 3×3, 128] ×2	[3×3, 128; 3×3, 128] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×8
conv4.x	14×14	[3×3, 256; 3×3, 256] ×2	[3×3, 256; 3×3, 256] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×23	[1×1, 256; 3×3, 256; 1×1, 1024] ×36
conv5.x	7×7	[3×3, 512; 3×3, 512] ×2	[3×3, 512; 3×3, 512] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 2. ResNet50 ARCHITECTURE DIAGRAM

A sample reference can be seen here [**?**].

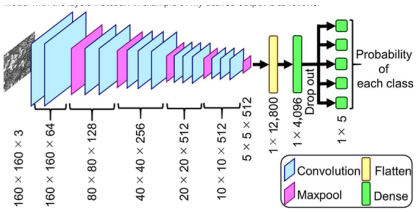

Figure 3. CNN MODEL ARCHITECTURE DIAGRAM

3 Experiments and results

We used our own dataset to train the pre-trained ResNet50 model. In order to train the data, we created a function that accepts the ResNet50 pre-trained model as one of its parameters. The CNN ResNet50 image classification model was trained on our dataset using the following tools and strategies: • Dataset :- For our dataset, we started by collecting 18000 images. Following that, the data was split into training and testing sets, with 3000 and 15,000 photographs, respectively, being utilised in each. • Data Preprocessing :- To make the dataset acceptable for training the model, the dataset's images must be pre-processed. With the help of the Keras tool "ImageDataGenerator," we pre-process and enhance the images. The training/testing dataset's final images are all 128*128 pixels in size. • Training :- We construct the model, provide the gradient descent, optimization methods, and prediction metrics before we can train it. The compile() method is used to compile the model, which takes the following arguments: model (optimizer, loss, metric, loss weights, sampling weight mode, weighting metrics, target tensors). • After that, we trained our model using the Keras library's.fit function. Model fitting refers to training the model using the training dataset.[7] • Testing :- Then the model was tested on the testing dataset using the .predict method from the Keras package. Just after model has been trained with 15000 photos, it is tested with 3000 new images to evaluate its efficiency. • Evaluation :- To verify the model's effectiveness, we utilised the Keras library's .evaluate function. Two measures were used to assess the model. Accuracy measures how well the model worked, while loss measures how bad the model predicted. • The VGG16 model is imported using the Keras library, and this is the result evaluation we get after running the code. [3]

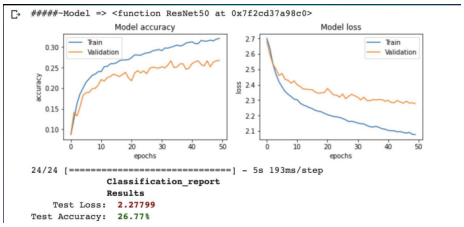

Figure 4. RESNET50 ACCURACY AND LOSS TABLE

[8]

4 Discussion

After testing all four models which are MobileNet , ResNet50 , VGG16, EfficientNetB7. We evaluated the models on two metrics- 1. Accuracy(How well the model)

2. Loss (How bad the model's prediction was) To compare the models we made the result table as well as graph.As we got MobileNet has highest accuracy of 0.47 and EfficientNetB7 has low-

	Model	Accuracy
0	VGG16	0.475667
1	ResNet50	0.267667
2	MobileNet	0.636667
3	EfficientNetB7	0.066667

Figure 5. ACCURACY OF EACH MODEL

est accuracy of 0.07. Also we have created a graph for Loss Result Table.In which EfficientNet has highest loss of 2.71.MobileNet has least loss of 1.23.

An N x N matrix called a "Confusion matrix," where N is the number of target classes, is used to assess the effectiveness of a classification model. The machine learning model's projected values are compared to the actual target values in the matrix.

Confusion Matrix is used to evaluate the performance of Machine Learning Models.Confusion Matrix shows us how many false positives, false negatives, true positives and true negatives you have. With the help of these you can calculate the two matrix:

1. Precision
2. Recall

Confusion matrix for MobileNet model

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	[2,	4,	0,	0,	1,	0,	0,	1,	2,	0,	1,	2,	0,	0,	2]
2	[0,	3,	0,	0,	2,	0,	0,	2,	0,	0,	2,	1,	0,	0,	5]
3	[3,	1,	1,	0,	0,	0,	1,	2,	0,	0,	6,	0,	0,	0,	1]
4	[0,	5,	0,	3,	0,	0,	0,	1,	0,	0,	4,	0,	0,	0,	2]
5	[5,	1,	0,	0,	0,	0,	0,	4,	2,	0,	0,	1,	0,	0,	2]
6	[2,	3,	0,	0,	0,	0,	0,	0,	3,	0,	0,	1,	0,	0,	6]
7	[1,	2,	0,	0,	0,	0,	0,	3,	1,	0,	4,	0,	0,	0,	4]
8	[0,	0,	0,	0,	2,	0,	0,	3,	1,	0,	2,	4,	0,	1,	2]
9	[3,	0,	0,	0,	2,	0,	0,	1,	1,	0,	2,	2,	0,	0,	4]
10	[4,	2,	0,	0,	0,	0,	0,	1,	3,	0,	4,	0,	0,	1,	0]
11	[1,	4,	0,	3,	0,	0,	0,	0,	1,	1,	5,	0,	0,	0,	0]
12	[1,	1,	0,	0,	0,	0,	0,	4,	1,	0,	2,	1,	0,	0,	5]
13	[1,	1,	0,	0,	2,	0,	2,	0,	4,	0,	1,	1,	0,	0,	3]
14	[2,	1,	0,	0,	1,	0,	1,	2,	1,	0,	0,	5,	0,	0,	2]
15	[1,	3,	0,	0,	3,	0,	0,	2,	0,	0,	1,	2,	0,	0,	3]

Figure 6. Confusion Matrix

5 Conclusion and future work

Mobile Net is proven to be the best model for our dataset with the highest accuracy and lowest loss rate.VGG16 is the second-best model for this dataset, followed by ResNet 50 and EffcientNetB7.We decided to implement some changes to the code and changing some parameters to better understand how it is working and also creating a confusion matrix.In future we would evaluate other models on the same dataset to see which works the best, we would also add to the dataset and see it other models would respond better training data. The experimental results show that this method has good accuracy and can effectively improve the accuracy of image recognition.

ACKNOWLEDGEMENTS

I would want to thank everyone who contributed to the success of our project. First of all, I want to thank Mr Mohammad Majid al-Rifaie our computer science instructor, who helped me much and directed me throughout the assignment. He helped us comprehend how to finish this project effectively, and without him, the project would not have been finished.

This project has helped us learn and apply our academic understanding to the actual world. I would want to thank him for his assistance and direction with this endeavour.

Additionally, I want to thank my parents for always being there for me. At last but not in least, we would like to thank everyone who helped and motivated us to work on this project.

REFERENCES

[1] Resnet50, Dec 2022.

[2] N. Archana and K. Hareesh, 'Real-time human activity recognition using resnet and 3d convolutional neural networks', *2021 2nd International Conference on Advances in Computing, Communication, Embedded and Secure Systems (ACCESS)*, (2021).

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, 'Deep residual learning for image recognition', in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, (2016).

[4] Riaz Ullah Khan, Xiaosong Zhang, Rajesh Kumar, and Emelia Opoku Aboagye, 'Evaluating the performance of resnet model based on image recognition', *Proceedings of the 2018 International Conference on Computing and Artificial Intelligence - ICCAI 2018*, (2018).

[5] Yixue Lin, Wanda Chi, Wenxue Sun, Shicai Liu, and Di Fan, 'Human action recognition algorithm based on improved resnet and skeletal keypoints in single image', *Mathematical Problems in Engineering*, **2020**, 1–12, (2020).

[6] Nattapon Noorit, Nikom Suvonvorn, and Montri Karnchanadecha, 'Model-based human action recognition', *SPIE Proceedings*, (2010).

[7] Michalis Vrigkas, Christophoros Nikou, and Ioannis A. Kakadiaris, 'A review of human activity recognition methods', *Frontiers in Robotics and AI*, **2**, (2015).

[8] Tian Yang, 'Research on human action recognition algorithm based on video image', *Online Journal of Robotics amp; Automation Technology*, **1**(2), (2021).