Hip Fracture Detection Using Deep Learning Approach

Or Benson
Omri Plotnik

Supervised by: Dana Cohen
Lecturer: Raja Giryes

University of Tel Aviv
or@mail.tau.ac.il
omriplot@gmail.com

October 20, 2022

Abstract

The primary function of the hip joint is to provide dynamic support to the weight of the body/trunk while facilitating force and load transmission from the axial skeleton to the lower extremities, allowing mobility. In addition to movement, the hip joint facilitates weight-bearing. Hip fractures can have serious consequences and therefore it is important to detect the fracture correctly and choose the right means of treatment. [Gold, Munjal, Varacallo, 2021]

In this work we tackle the problem of using small data to train nueral networks. We used different deep learning models to find out which one correctly detect the fractures. One of the models was a self-supervised one in order to deal with the small data. We also implemented GAN do increase our data with new instances similar to our orriginal data.

I. Introduction

A hip fracture is a partial or complete break of the femur (thigh bone), where it meets the pelvic bone.It's a serious injury that requires immediate medical attention. The majority of hip fractures happen to people over 60. In older patients, these include: a greater chance of death, a challenging recovery, diminished quality of life, depression and more. [Johns Hopkins medicine]

In order to diagnose the fractured bone, the doctor uses x – ray image. During this project we worked with Prof. Heller from Belinson hospital who works at the orthopedic department. According to him 5%-10% of surgeries fail due to wrong detection and classification of the fracture.

In this paper we implemented different CNN architectures to find out which one mange to correctly detect the location of the fracture. One of the main problems of working with medical data is that sometimes it is hard to have access to medical images and our dataset is small. In order to train neural networks and received good results we need hundreds of images. In this project our dataset only had 96 images and we applied different methods such as image augmentation, self-supervised models and GAN do deal with this problem.

II. Background

There are three broad categories of hip fractures based on the location of the fracture: femoral neck fractures, intertrochanteric frac-

tures, and subtrochanteric fractures. The femoral neck is the most common location for a hip fracture, accounting for 45% to 53% of hip fractures.

The femoral neck is the region of the femur bounded by the femoral head proximally and the greater and lesser trochanters distally. A femoral neck fracture is intracapsular, that is within the hip joint and beneath the fibrous joint capsule.

Figure 1: *Femoral neck fractures area*

Intertrochanteric fractures are breaks of the femur between the greater and the lesser trochanters. They are extracapsular fractures that is, outside the hip joint's fibrous capsule.

Figure 2: *Intertrochanteric fractures area*

Subtrochanteric fractures are located between the lesser trochanter and the femoral isthmus that is, in the proximal part of the femoral shaft. They are less common than femoral neck and intertrochanteric fractures, accounting for approximately 5% to 15% of hip fractures. Subtrochanteric fractures are less stable than the other two types of hip fractures and, consequently, more difficult to fix. [Simon Mears]

Figure 3: *Subtrochanteric fractures area*

III. Related Work

The main reference paper we implemented is "Bone Fracture Detection and Classification using Deep Learning Approach". [Yadav, Dhirendra & Rathor, Sandeep, 2020]. In this work, deep neural network was developed for the identification and the classification of the healthy and the fractured bone. Using different activation functions and different distribution of the images to training and test groups the best accuracy that has been achieved in this research was 95.67%.

i. MOCO

Momentum Contrast for unsupervised visual representation learning: Traditionally computer vision models have been trained using supervised learning. Different labels were created for images so that the model could learn the patterns of those labels. In self – supervised learning however, the model learns from unlabeled sample data. We apply different augmentations: resize, rotation and horizontal flip on the images and we let the model learn that these images still contain the same visual information.

In MoCo the unsupervised learning process is framed as a dictionary look – up. Each image is assigned a key. The key is generated by encoding each image using a CNN with the output being a vector representation of the image. When a new image enters MoCo, it is encoded into a vector representation and

will belong to one of the keys in the dictionary with the lowest distance.[Leon Sick, 2022], [He, Kaiming, et al, 2020]

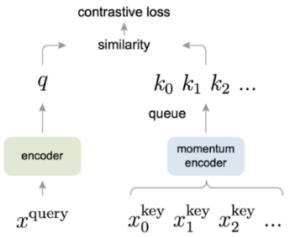

Figure 4: *MoCo process*

The training object is formulated by the InfoNCE loss function [He, Kaiming, et al, 2020]:

$$L_q = -log\frac{exp(qk_+/\tau)}{\sum_{i=0}^{K} exp(qk_i/\tau)} \quad (1)$$

ii. GAN

Generative Adversarial Network. This network comprised of two neural networks (Generator and Discriminator) that together create new data instances that resemble the training data. The Generator is used to generate the image from the random distribution of data to make the distribution D' to be close to the real images distribution D. The Discriminator discriminates between two different classes of data, the sample data and the generative one. [Amit Chauhan, 2021]

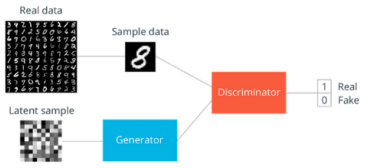

Figure 5: *GAN architecture*

The weights and biases of the model in the generator and discriminator are being updated as long as the discriminator distinguishes between the fake and real image. The formula for loss function is Binary cross – entropy [Amit Chauhan, 2021]:

$$L(y, \hat{y}) = [ylog(\hat{y}) + (1-y)log(1-\hat{y})] \quad (2)$$

Where y-hat is a reconstructed image and y is an original image

IV. Proposed Work

i. Obtaining the data

The training and testing of our neural network are based on pictures of hip fractures. For this project we collaborated with Prof. Heller and received 96 pictures of hips separated to two categories: left (50 images) and right (46 images) in accordance with the position of the fracture. All images include interrochanteric fractures only. We couldn't obtain images with other type of fractures.

Figure 6: *An example of the data*

ii. Increasing the image database

As pointed before we received only 96 pictures of hips. When we train a deep learning model, we in fact tuning its parameters such that it can map a particular input (in our case an image) to some output (a label). If we have a lot of parameters, we would need to show our model a proportional amount of examples, to get good performance. The number of parameters we need is proportional to the complexity of the task our model has to perform.[Arun Gandhi, 2021]

We couldn't receive more images and therefore had to find other ways to increase our dataset. First, since the hips are symmetrical,

we split each picture to two parts, left and right and labeled each part, with or without a fracture. Second, we used image augmentation to further enlarge the dataset. We implemented image resize, horizontal flip and rotation.

iii. Methods

In the paper that we chose the researchers concluded that "The accuracy of the model can be further improved by selection of other deep learning model". [Yadav, Dhirendra & Rathor, Sandeep, 2020] Therefore, in this work we decided to compare different CNN models. Each network was trained and tested on the same data. The models we trained are:

- Vgg16
- Vgg19
- Resnet9
- Resnet34
- efficiNet
- mobileNet
- MoCo, a self supervised model
- GAN

V. Results

Comparing the different models we get the results as seen in Figure 7. We can see that efficientNet and mobileNet performed the best.

Name\Categories	Sensitivity	Specificity	Precision	Accuracy	AUC
VGG16	0.628	0.893	0.695	0.749	0.494
VGG19	0.604	**0.914**	0.702	0.497	0.533
ResNet9	0.782	0.794	0.773	0.909	0.444
ResNet34	0.766	0.785	0.763	0.883	0.495
EfficientNet	0.885	0.896	0.802	**0.993**	0.953
Mobile	**0.979**	0.883	0.811	0.838	**0.996**
MoCo	0.813	0.807	**0.850**	0.767	0.958

Figure 7: *The models results*

In appendices 3-5 there are the confusion matrices and loss and accuracy graphs used to calculate the parameters above.

We also calculate the ROC curves for the models as can be seen in Figure 8. The ROC curve tell us the trade-off between sensitivity and specificity. Models that are closer to the top - left corner perform better. We got that Mobilenet has the highest AUC and sensitivity.

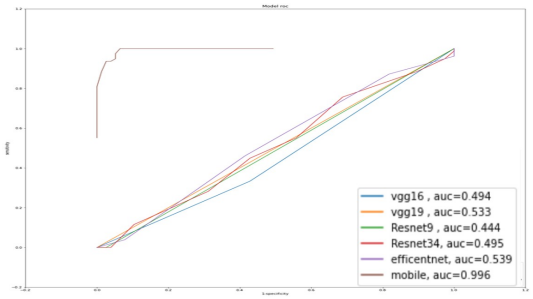

Figure 8: *The ROC curves of the different models*

We create a Grad-Cam for each model. This image represent the gradients of the classification score which helps us identify the parts of an input image that most impact the classification score. We can see from Figure 9 that Mobilenet was able to correctly detect the location of the fracture. On the other hand we can see from Figure 10 that Resent9, for example, didn't detect any fracture. (All Grad – Cam images are in Appendix 2)

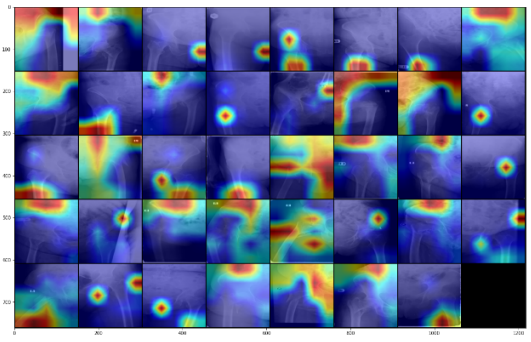

Figure 9: *mobileNet Grad - Cam*

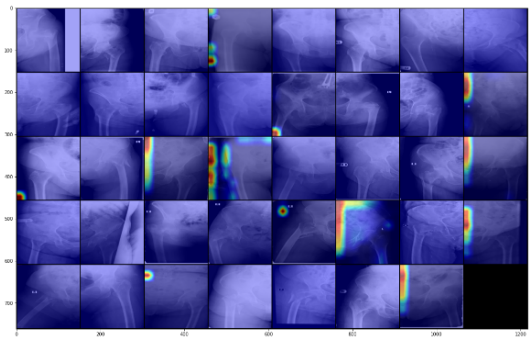

Figure 10: *resnet9 Grad - Cam*

For the GAN we used 500 epochs and received new data instances that resemble our training data as can be seen in Figure 11.

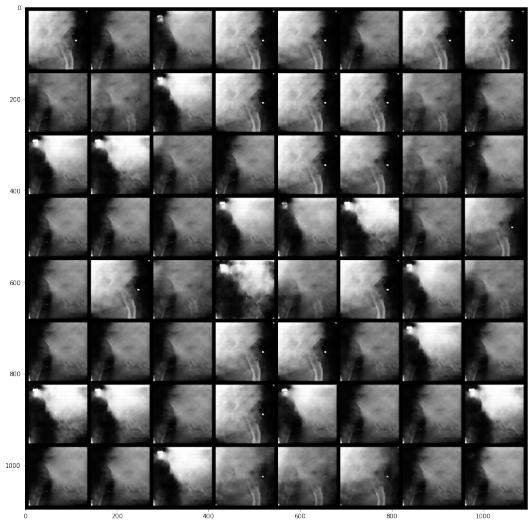

Figure 11: *GAN new data*

We got results only for Resnet9 and MobileNet as can be seen from Figure 12. We can see that by adding the new images from the GAN to the original data, we received higher accuracy in both models.

VI. Conclusion

In this paper we used different models and methods to detect hip fractures from small dataset of X - ray images. Our data contained around 100 images, and it was augmented to deal with working with small data. The model with the best AUC was mobileNet with 0.996. Using the GAN to generate new data instances increased the accuracy of the models we used the data on. Our highest accuracy model was efficentNet with 99.3% compare to 95.67% in the reference paper.

For further research more data is needed. Moreover, in this paper we only had one type of hip fractures images. In future work, with different types of hip fractures we can also check the performance of the models we used for classification of hip fractures.

Name\Categories	Sensitivity	Specificity	Precision	Accuracy	AUC
ResNet9	0.604	**0.914**	0.702	0.497	0.533
ResNet9+Gan	0.705	0.794	0.744	0.672	0.608
Mobile	**0.979**	0.883	0.811	0.838	**0.996**
Mobile+Gan	0.944	0.825	**0.836**	**0.872**	0.942

Figure 12: *The models results after GAN*

References

[Gold, Munjal, Varacallo, 2021] Gold M, Munjal A, Varacallo M. Anatomy. (2021 Jul 31) Bony Pelvis and Lower Limb, Hip Joint.

[Johns Hopkins medicine] `https://www.hopkinsmedicine.org/health/conditions-and-diseases/hip-fracture`

[Simon Mears] Simon Mears, MD. `https://www.hopkinsmedicine.org/gec/series/fixing_hip_fractures#intertrochanteric_fractures`

[Yadav, Dhirendra & Rathor, Sandeep, 2020] Yadav, Dhirendra & Rathor, Sandeep. (2020). "Bone Fracture Detection and Classification using Deep Learning Approach"

[Leon Sick, 2022] Leon Sick, 2022. Paper explained: Momentum Contrast for Unsupervised Visual Representation Learning

[He, Kaiming, et al, 2020] He, Kaiming, et al. (2020) "Momentum contrast for unsupervised visual representation learning".

[Amit Chauhan, 2021] Amit Chauhan. (2021) "Understand Generative Adversarial Network (GAN) in Deep Learning".

[Arun Gandhi, 2021] Arun Gandhi. (2021) "Data Augmentation | How to use Deep Learning when you have Limited Data — Part 2".

VII. Appendix

i. Appendix 1 - Our code

Our code can be found at drive: `https://drive.google.com/drive/folders/1RTGAOpSvW-PEEjOwTY3-Ar5a3mAPiEmi?usp=sharing`

ii. Appendix 2 - Grad Cam images

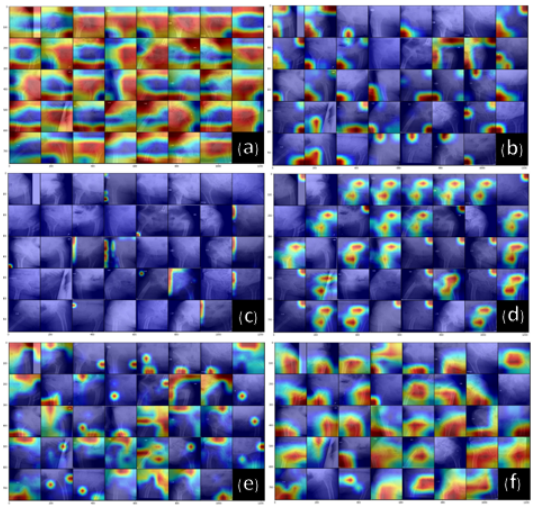

Figure 13: *(a) Vgg16 , (b) Vgg19 , (c) resnet9 , (d) resnet34 , (e) mobileNet , (f) efficientNet*

iii. Appendix 3 - Confusion matrix

The confusion matrices of our models:

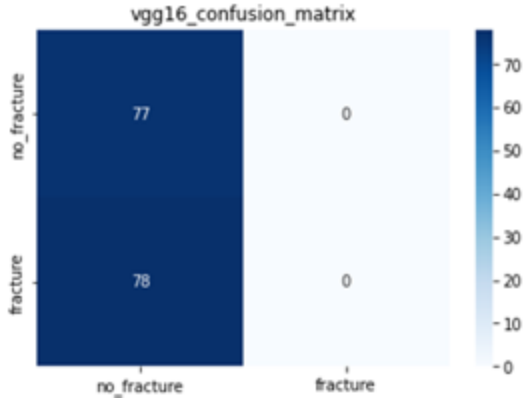

Figure 14: *Vgg16 confusion matrix*

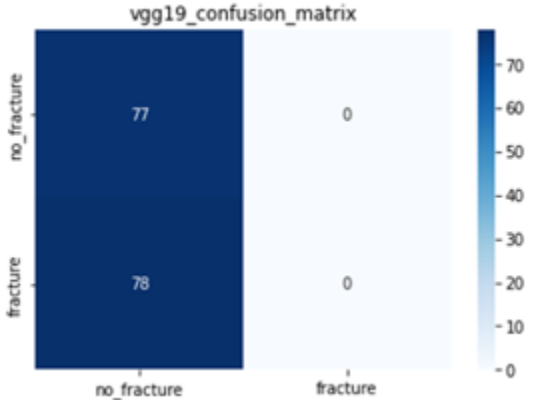

Figure 15: *Vgg19 confusion matrix*

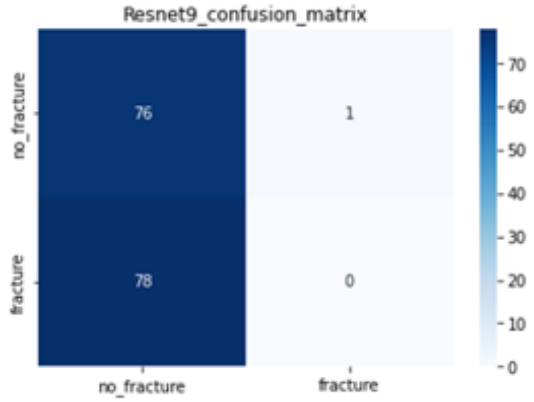

Figure 16: *ResNet9 confusion matrix*

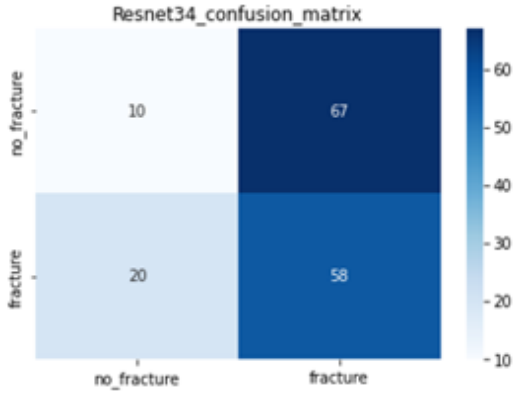

Figure 17: *ResNet34 confusion matrix*

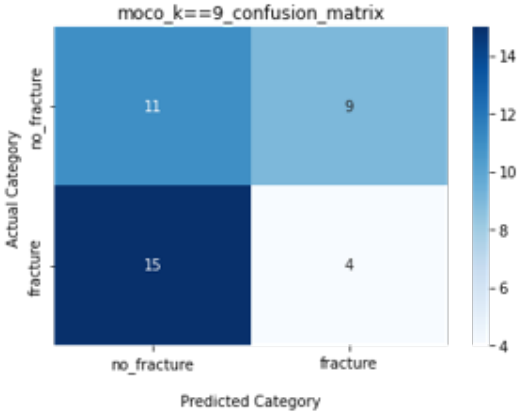

Figure 20: *MOCO, self - supervised model confusion matrix*

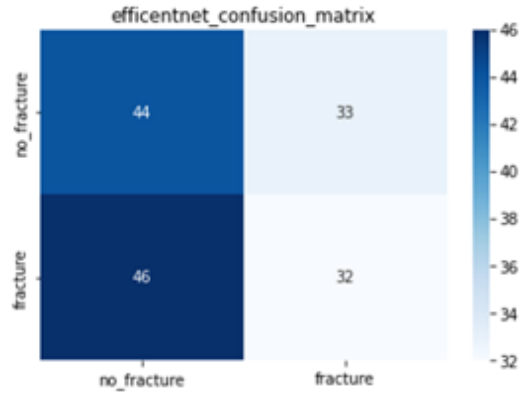

Figure 18: *efficentNet confusion matrix*

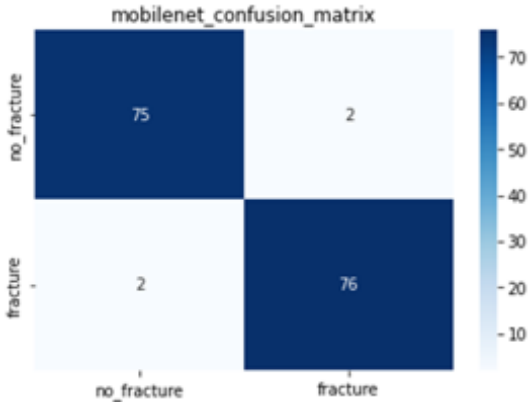

Figure 19: *mobileNet confusion matrix*

iv. Appendix 4 - Model Loss

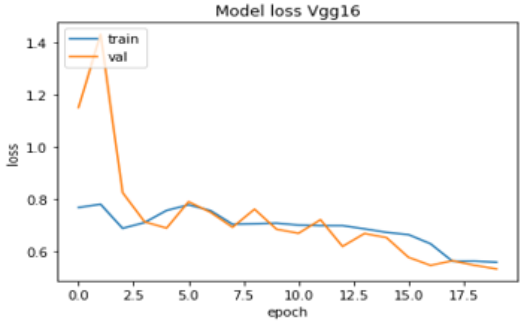

Figure 21: *Vgg16 loss*

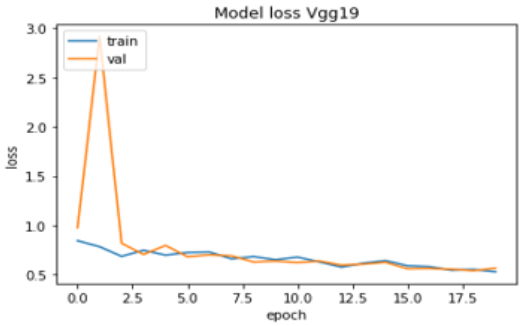

Figure 22: *Vgg19 loss*

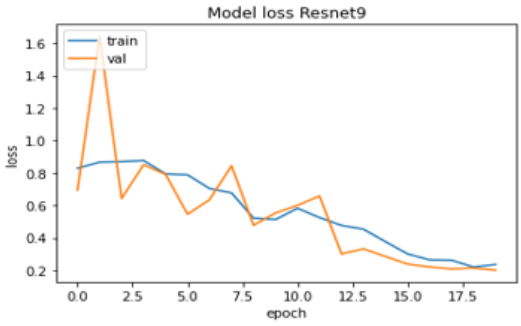

Figure 23: *ResNet9 loss*

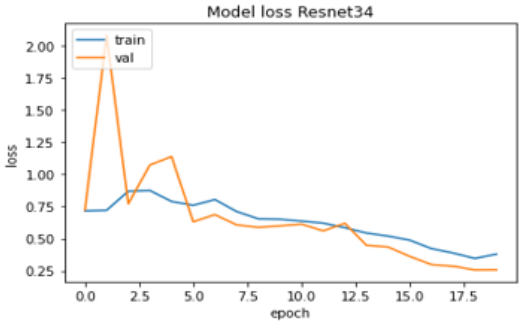

Figure 24: *ResNet34 loss*

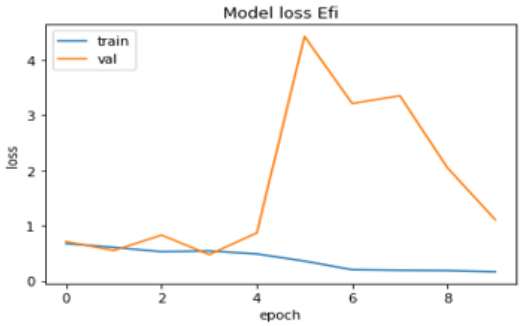

Figure 25: *efficentNet loss*

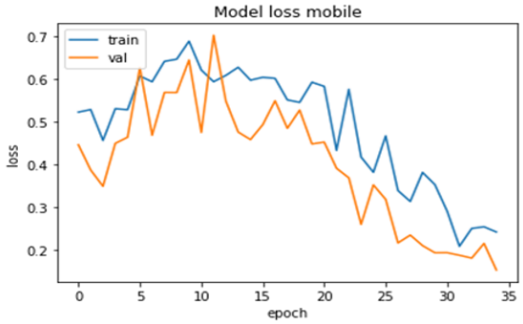

Figure 26: *mobileNet loss*

v. Appendix 5 - Model accuracy

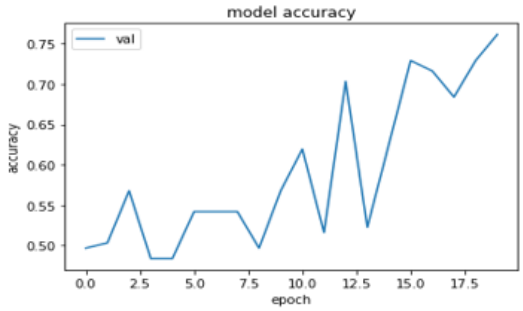

Figure 27: *Vgg16 accuracy*

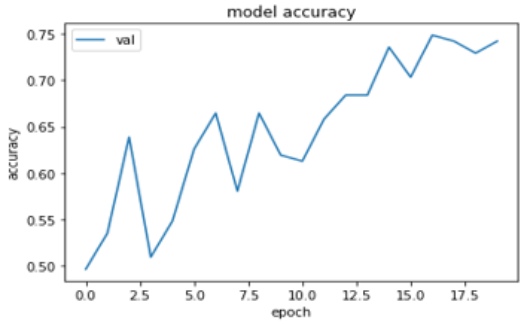

Figure 28: *Vgg19 accuracy*

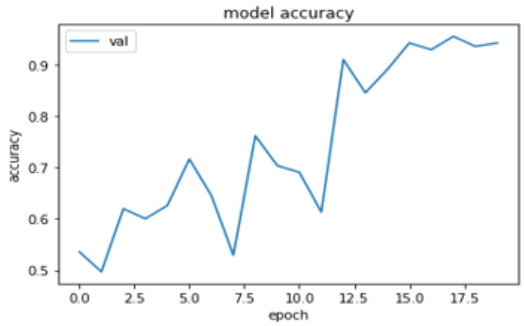

Figure 29: *Resnet9 accuracy*

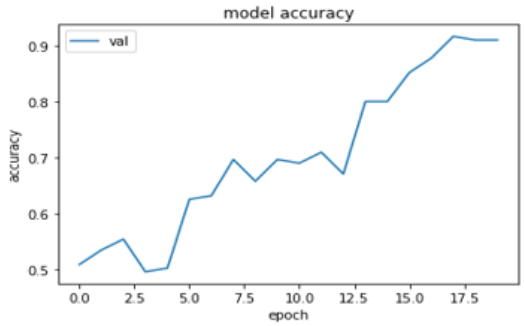

Figure 30: *Resnet34 accuracy*

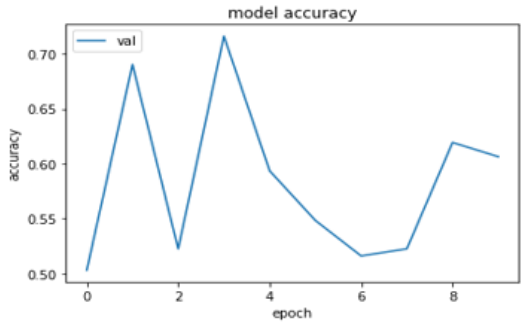

Figure 31: *efficientNet accuracy*

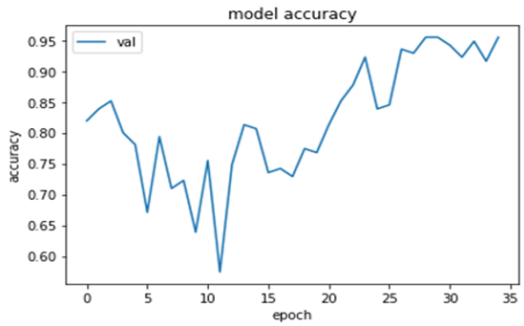

Figure 32: *mobileNet accuracy*