Bus Body Fitness Detection Using CNN

Md. Jadid Hossain Sanim
Dept. of

Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
jadid.hossain.sanim@g.bracu.ac.bd

Md. Tasnimul Hassan Khan Rafath
Dept. of

Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
tasnimul.hassan.khan@g.bracu.ac.bd

Muhammad Tasib Ahasan
Dept. of

Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
muhammad.tasib.ahasan@g.bracu.ac.bd

Abstract—**Road safety is one of the major issues in Bangladesh. Here, the huge population daily uses buses as their primary transportation. But there are many buses here that are unfit to drive, which is causing serious accidents and increasing the risk of casualties. But due to the lack of proper strictness of the government and absence of efficient detection methods unfit buses cannot be stopped. So, here we will use deep learning models like Convolution Neural Network (CNN) to identify fit & unfit buses and take proper action. Popular pre-trained models such as VGG16, ResNet50, EfficientNetB0, InceptionV3 and DenseNet-121 are used here to identify fit & unfit buses with more accuracy and precision. Among them the custom CNN model provided an accuracy of 88.67% & EfficientNetB0 model obtained the highest accuracy of 95.13% which is the highest among all the models. This gives us the potential to improve bus body fitness monitoring and opens different ways to further research in the deep learning field.**

Index Terms— **Bus Body Fitness Detection, Convolutional Neural Networks, VCG16, EfficientNetB0, ResNet50, DenseNet-121, Computer Vision, Road Safety, Deep Learning.**

I. INTRODUCTION

Road safety is a major issue worldwide. It has become a widespread problem in many countries. Many people die in road accidents all over the world. Even in developed countries like the United States, where there are strict traffic laws, 15,000 bus accidents occur every year [1]. But the situation in underdeveloped countries is even worse. According to the World Health Organization, 92% of the world's road deaths occur in low- and middle-income countries. [2]. Bangladesh is one of them. Bangladesh is a rapidly developing country with a large and densely populated population. This large developing population needs easy and cheap transportation. Therefore, the role of buses as public transport in this country is increasing. As a result, the number of various types of buses is also increasing. However one of the challenges of this need is the issue of bus body fitness, which directly impacts public safety and the efficiency of the transportation network.

Bus body fitness refers to the structural condition of a bus, ensuring that it meets the country's safety and emission standards so that it does not cause any accidents. But road accidents are increasing at an alarming rate in Bangladesh. Around 25,000 people die in road accidents every year[3]. Unfit buses are one of the main causes, directly or indirectly. According to a report by the Daily Star, 21.92 percent of buses in Dhaka city alone do not have any fitness clearances, and 1,600 buses are being operated without any authorization and fitness.[4] And according to the data

published by BRTA, that number is increasing every year [5]. In many cases, the bus body is not repaired or changed but is repainted and put back on the road. According to ARI Director Prof Md Hadiuzzaman, "Owners of unfit vehicles have hardly taken buses or cars to workshops for maintenance, so many of these vehicles are past their economic life" [5], as a result of which all passengers have to travel on the road with the risk of death. On the other hand, the massive black smoke from these unfit buses has become one of the main causes of environmental and air pollution in Bangladesh. This has created a terrible health risk for people. According to research, unfit vehicles are responsible for 15% of Dhaka's air pollution [6].

The traffic police are mainly responsible for overcoming this problem. But due to their negligence and lack of proper strictness by the government, unfit buses cannot be stopped. There are many examples where the police take bribes from unfit buses by trading cases and allowing them to move freely. Again, there is not enough police or manpower to maintain such a populated landscape. Therefore, advanced technology like deep learning & computer vision can be used to identify these unfit buses and take action. For this research we used the guidelines from BRTA to label the buses as fit and unfit. The guidelines included checking headlights, glass, rearview mirror, indicator, body crack, body scratch, etc. For this detection, we mainly used a Convolution Neural Network (CNN). CNN models can take images of buses and classify them into 2 categories: fit and unfit. For this reason, we trained the model with a dataset of almost 3000 images into two types of categories: fit buses and unfit buses. We took some raw image data from different places of Dhaka and other image data from online credible sources such as open source platform Kaggle. The main objective of our research is to build a deep learning model that can automatically classify bus images and help the government take appropriate action to mitigate the road accidents.

II. LITERATURE REVIEW

There is a lot of research on the fitness detection of different vehicles. However, only one research has been found specifically on bus body fitness. Bus body fitness is determined through damage part detection of the bus body. Therefore, these studies have been given more importance in this research.

Khan et al. (2022) [7] showed a real-time method for detecting the fitness of buses. He proposed to use YOLOv5 named deep learning object detection model. His paper tried to identify unfit buses by analyzing some key body

components like wheels, steering, body parts, scratches, broken glass, headlights, etc. After preprocessing his raw data, he used the YOLOv5 model for training and testing. That model helped to achieve 97.3% accuracy across all classes. The model showed the potential for vehicle fitness monitoring. However, this research could not be used for other vehicle types and different scenarios.

Zhu et al. (2021) [8] discuss the use of deep learning-based framework for vehicle damage recognition. It uses Mask R-CNN and is enhanced with KL-loss to improve the precision of damage localization. The study showed a way to resolve ambiguous damage scopes, that is the upgrade of traditional IOU-based metrics by introducing a component assisted evaluation mechanism. This research provides an accuracy of 81.6% and the recall of the model is 67.4%. The study paves the way for automated, intelligent solutions in vehicle damage assessment, with the potential for further advancements in accuracy and commercial applications. However, this research suggested combining both Mask R-CNN and Yolov8 to find more efficient and accurate output.

Zhang et al. (2020) [9] focuses on the improved MobileNet-SSD algorithm for vehicle paint defect detection. Sometimes when the model is run on a smaller dataset, its accuracy is overfitted. For this reason, we cannot get the correct idea of the desired output. Here a custom MobileNet-SSD is proposed instead of the basic SSD and VGG16. And it is 10% faster and 95% more accurate than the common algorithm. However, one limitation of this model is that it cannot detect small objects properly. Also, this research has proposed to use K-means clustering approach to get more accuracy through exact boundary box detection.

Azmi et al. (2022) [10] proposed a model that helps to detect damage of the vehicle based on the YOLOv4 model. They used LabelIng to identify the four damage types: dents, broken glass, scratches, and damaged tail light of the vehicle body. Also applied data augmentation techniques. This model got an accuracy of 81.20%. This paper also used CSPDarknet53 for that.

III. METHODOLOGY

Dataset pre-processing is one of the most important steps in machine learning. Through this, a dataset is basically prepared for training and testing a model. The better the pre-processing, the better the model will learn and the higher the accuracy will be. Through this, we remove unwanted data and keep only the necessary data. Again, the data is further improved by various techniques on that necessary data. For different types of data, different types of pre-processing techniques will be followed. So that unwanted data will not learn, and it will take less time and be efficient.

A. Dataset Description:

We will mainly work with image data in this research. We collected datasets in two ways. The first is through various trusted online dataset platforms such as Mendley, Kaggle. From these 2 sources, we collected approximately 5000 data. The second is that we manually collected raw data from the field ourselves. The number of these raw data is approximately 750. This collection has various types of buses of different heights and widths. The Images were shot in various sunlight conditions. This data was divided in two labels into two folders: Fit and unfit for training & testing. Then our pre-processing part begins.

B. Data Preprocessing:

The preprocessing of the image dataset was performed to enhance model generalization. It helped to prepare the dataset for effective training and testing. Firstly, We cleaned the unnecessary and bad quality images from the dataset manually . Then we removed the duplicate images. In this way, our total dataset of 5750 images Shrinked to a dataset of 2870 images. Then extensive augmentation was applied to increase the diversity of the dataset and to bring in various scenarios. This included first normalizing all the data through rescaling. Then all the data was randomly zoomed up to 30%. After that, it was flipped horizontally and rotated up to 20 degrees. Then the augmentation was completed by adjusting the RGB brightness. Then the second step was to convert all these images to 128x128 pixels and create a batch of 32 images at the same time. After that the image preprocessing is completed.
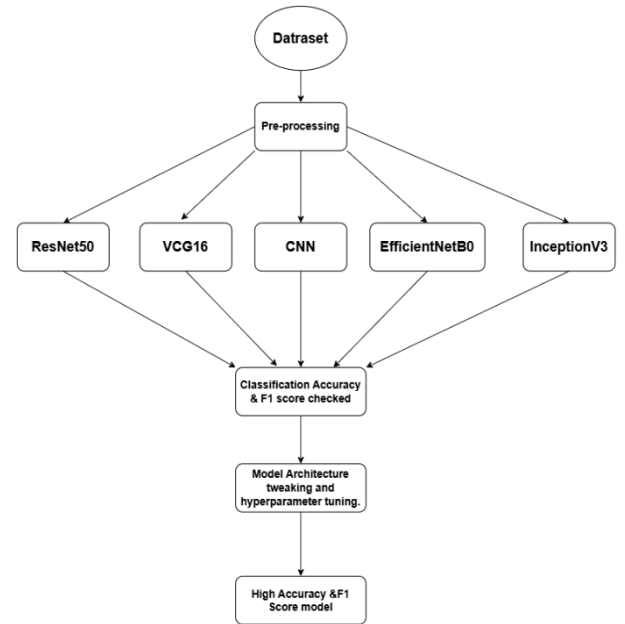

Fig. 1: Flow Chart of the Bus Body Fitness Detection model.

C. Learning Phase:

1.Custom CNN: Convolutional Neural Network (CNN) is a deep learning model for tasks like spatial data, images, and videos. It has attempted to provide higher accuracy in all fields that it has employed like object detection, digit, and image recognition [11]. In this model convolutional layers are used to find some special features that help to make feature maps. Then max pooling is used to make it more efficient. In this research, we tested 3 different architectures of Basic CNN and the last architecture gave the best result which was found through extensive manual testing. Unlike traditional CNN models, our CNN model uses a deeper architecture with a smaller (3x3) kernel size,

systematically using MaxPooling and progressively increasing filters(16-128). We used modern enhancements like Dropout to reduce overfitting, L2 regularization for better abstraction, Batch Normalization for stable training, and Adam Optimizer for better convergence.

2.VCG16: VCG16 is a CNN model which is also known as ConvNet is one of the most popular CNN models that is used in image classifications and is easy to use with the help of transfer learning. The model consists of 16 layers, Including fully connected 3 layers, 13 convolutional layers, and 5 max pooling layers [12]. Each layer is organized into blocks and each of them contains multiple convolutional layers with a max-pooling layer for down sampling.

3.ResNet50: It is a CNN-based architecture that belongs to the ResNet family. It is developed by the researchers of Microsoft Research Asia and is known for its efficiency and depth in Image classification problems. RestNet50 architecture has different depths of variations like ResNet-32, ResNet-18, and so on. ResNet50 consists of 50 weight layers and it has an image input size of 224-by-224 [14].

4.EfficientNetB0: It is a Convolution Neural Network that was trained on more than a million images of the ImageNet database. It is the most efficient and smallest model in the EfficientNet family. Its layers are based on a compound scaling method that scales the resolution, width, and depth of the network. In the case of EfficientNetB0 architecture, it consists of Batch Normalization, stride 2 with 3x3 convolution and Swish activation [15].

5.InceptionV3: Inception V3 is a pre-trained CNN model with 48 layers built by Google. It is known for its low computational cost and high efficiency in image recognition. It was trained on more than one million images and it can classify images into 1000 different categories such as animals, pencils etc. It has an image input size of 299-by-299. The model extracts general features from input images in the first part and classifies them based on those features in the second part [13]. It is also used in feature extraction and transfer learning.

6.DenseNet-121: DenseNet-121 is a neural network introduced by Laurens van der Maaten, Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. It stands for Densely Connected Convolutional Networks and is known for its accuracy and efficiency in image-related classification tasks. In a feedforward fashion it connects every layer to other layers. DenseNet-121 uses bottleneck layers which reduces the number of parameters without reducing the feature numbers of the network [16].

IV. RESULTS AND ANALYSIS

The objective of this study was to develop a deep learning model that can classify bus images as either "Fit" or "Unfit". In our research, we employed a custom convolutional neural network (CNN) alongside the widely used and tested pre-trained models VGG16, ResNet50, EfficientNetB0, and InceptionV3. Subsequently, different tests were carried out to examine the performance of these models as well as fine-tuning the custom CNN model by altering its architecture and fine-tuning its hyperparameters. The models were tested thoroughly to see how well they performed on new, unseen data. Their performance effectiveness was measured through accuracy, validation loss and F1-score.

1. **Custom CNN Model:**

The first custom CNN model was built with simple architecture that included filters of size 16, 32, and 64, but it didn't perform well. Another model was built with 32, 64, and 128 filters; however, it failed to capture all the features. Then the model architecture was developed using the optimized parameters found by automatic hyper-parameter tuning. Tuner RandomSearch was used for finding the most effective hyper-parameters that best fit the model. The use of hyperparameter tuning from RandomSearch established the best mix of filters and dropout rates to achieve the best results. These included: Filters=128, Dropout Rate=0.3, Dense Units=128, Learning Rate=0.001. Then the model is optimized using the Adam optimizer, with a learning rate of 0.001 and using binary cross-entropy loss function. This architecture gave the best performance among the 3 CNN architectures used on this dataset. The model reached an accuracy of 88.67% on the test set. While training, the accuracy kept improving and the model performed best with extensive manual hyperparameter tuning (learning rate, dropout rate, and dense layer size). This shows that with proper tuning, a simple custom CNN can achieve competitive results for binary classification tasks.
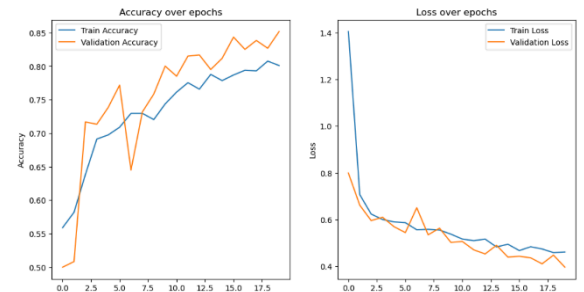
Fig. 2: Plot Training History of CNN Model.

2. **Transfer Learning with Pretrained Models:**

We used transfer learning here with some popular pre-trained models including VGG16, ResNet50, EfficientNetB0, InceptionV3 and DenseNet-121. The performances of these models varied in terms of accuracy and validation loss providing useful insight into how the structure of the model and transfer learning affect the classification task.

VGG16:

VGG16 represents a very deep architecture which is simple simultaneously. Indeed, it outperformed the custom CNN model in the classification of bus images. It showed excellent performance, especially when fine-tuning on the dataset. The model VGG16 gave an accuracy of 90.10% and

F1-Score of 0.90. It achieved high accuracy thanks to the quality of the ImageNet features that were used for initialization.

ResNet50:
Also doing very well is the ResNet50, having deep residual connections. It has outperformed VGG16 with quite a margin along with an accuracy of 93.50% & F1-Score of 0.93. The model showed the employment of deeper networks with skip connections in overcoming overfitting to allow model generalization.

EfficientNetB0:
EfficientNetB0 outperformed both VGG16 and ResNet50 to attain the highest accuracy. It achieved an accuracy of 95.13% and F1-Score of 0.94. It is the best model for our research so far. EfficientNetB0 performed better than other models on this dataset because of its efficient design which balances accuracy and computational efficiency. Its unique scaling method improves depth, width and resolution, making it great for extracting features in complex tasks like detecting bus body fitness. The model's pretrained weights, smart use of parameters and modern training methods also helped it perform well with less data and fewer resources. This model is efficient and ideal for image classification tasks.

DenseNet-121:
In addition, the most prominent feature of the DenseNet121 model is its connectivity pattern being dense, as every layer connects to all others. It also portrayed very strong performances. It resulted in an accuracy of 91.67% with an F1-Score of 0.91. Though efficient in using the parameters, it tends to have lower performance compared with some models such as EfficientNetB0, probably because of its complex architecture and the challenge of the problem to solve.
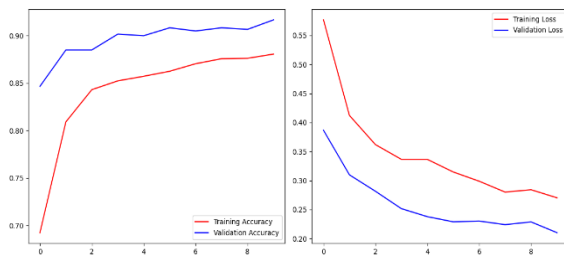

Fig. 3: Plot Training History of DenseNet-121 Model.

InceptionV3:
The other one is InceptionV3, noted for its multi-branch architecture with different kernel sizes; this architecture achieved competitive results, too, although a bit lower than other models. The model achieved an accuracy of 89% with an F1-Score of 0.89. It showed very similar results to ResNet50 but can be a little worse in this case due to its more complicated architecture.

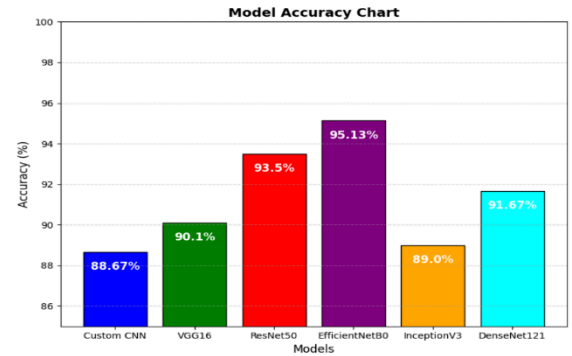Now, here is the visualization of model comparison.

Fig. 4: Bar chart of comparison of different models.

3. Confusion Matrix and Classification Report:
The confusion matrix and the classification report show the models performed reasonably well for classifying "Fit" and "Unfit" for the images. This is obvious from the confusion matrix that is given for models like custom CNN, where there are very few false positives and false negatives. Similarly, for all models, precision was high which means greater portions of positive predictions were correct. The Recall values were very high as it turned out, because the models could identify most of the positive cases of "Fit" or "Unfit" buses correctly. The average F1-score was always greater than 0.9 for the best models, which is a very good balance between precision and recall.

TABLE I
PERFORMANCE TABLE

Model	Performance Metrics	
	Accuracy (%)	F1 Score (%)
Custom CNN	88.67%	0.88
VCG16	90.1%	0.9
ResNet50	93.5%	0.93
EfficientNetB0	95.13%	0.94
InceptionV3	89%	0.89
DenseNet-121	91.67%	0.91

4. Feature Map Visualization:
Feature map visualization using VGG16 showed the strength of the model in learning complex features. It showed patterns in the first convolutional layer indicative of edges, textures and other basic image features. These visualizations helped confirm the hypothesis that models learn meaningful features that distinguish the classes "Fit" versus "Unfit".
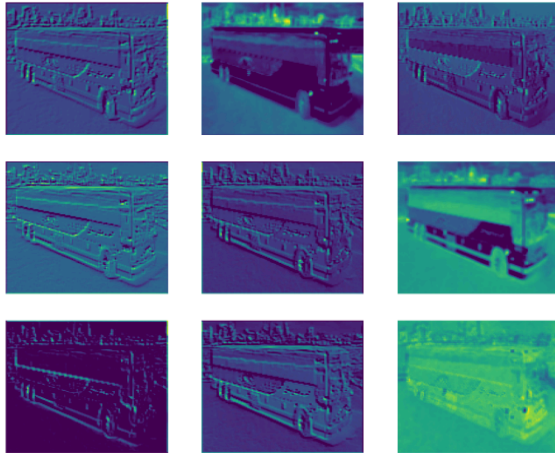
Fig. 5: Feature Map Visualization of CNN Model.

5. **Live Testing and Predictions:**

The test on new images proved that the models were good in making predictions, especially the EfficientNetB0 model. On the various test images, it outputted close-to-1 prediction scores for "Unfit" buses and close to 0 for "Fit" buses which upon visual inspection turned out correct. It confirmed that the models managed to generalize well on unseen data.

This study confirms that custom architecture CNNs with hyperparameter tuning may be efficient in image classification tasks such as the classification of buses into "Fit" or "Unfit" in this case. Among all the models EfficientNetB0 has become the best model and outperformed everyone considering accuracy. While custom CNN models perform well, fine-tuning EfficientNetB0 greatly increases both performance and efficiency and should be the preferred choice when performing real-world image classification problems.

V. CONCLUSION

The proposed Bus body fitness detection model has different kinds of limitations, but it provides a good amount of accuracy. Here, the research tried to explore and compare different models to combine and find the most accuracy in this field. However, the dataset was relatively small and data quality also was not very good. That's why it could not obtain the maximum accuracy. Our aim was to observe the body of the bus and make a decision of fitness based on the CNN models. The future scope of this research can be adding a raw, large, accurate and healthy dataset to train the model which can give the new scope of research. We believe that this proposed model can be more perfect and accurate if we increase the new larger and healthy dataset.

REFERENCES

[1] *Bus accident statistics – 2023 edition. TruckInfo.net. (n.d.). https://www.truckinfo.net/research/bus-accident-statistics*

[2] World Health Organization. (n.d.). *Road traffic injuries*. World Health Organization. https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries

[3] Rita, S. (2023a, October 22). 25,000 lives lost in road accidents every year. *Dhaka Tribune*. Retrieved January 6, 2025, from https://www.dhakatribune.com/.

[4] Adhikary, T. S. (2023a, June 9). Unfit buses rule city streets. The Daily Star. Retrieved January 6, 2025, from https://www.thedailystar.net/.

[5] Mamun, S. (2022, April 27). 2 million unlicensed drivers, half million unfit vehicles threatening road safety. Dhaka Tribune.

[6] Retrieved January 6, 2025, from https://www.dhakatribune.com/bangladesh/269196/2-million-unlicensed-drivers-half-million-unfit.

[7] Antara, N. F. (n.d.). Unfit vehicles responsible for 15% of air pollution in dhaka. DhakaTribune. Retrieved January 6, 2025, from https://www.dhakatribune.com/bangladesh/dhaka/259448/study-unfit-vehicles-responsible-for-15%25-of-air.

[8] M. M. Khan, D. M. Raza and M. A. H. Nahid, "Real-Time Vehicle fitness detection using YOLOv5," 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India, 2023, pp. 1-7, doi: 10.1109/ICCCNT56998.2023.10307679.

[9] Zhu, Q., Hu, W., Liu, Y., & Zhao, Z. (2021). Research on vehicle appearance damage recognition based on deep learning. Journal of Physics: Conference Series, 1880(1), 012024. https://doi.org/10.1088/1742-6596/1880/1/012024

[10] Zhang, J., Xu, J., Zhu, L., Zhang, K., Liu, T., Wang, D., & Wang, X. (2020). An improved MobileNet-SSD algorithm for automatic defect detection on vehicle body paint. Multimedia Tools and Applications, 79(31), 23367–23385. https://doi.org/10.1007/s11042-020-09152-6

[11] Azmi, M. A., Ahmad, I., Arefeen, M. Z., Ahmed, D., & Mehare, H. B. (2022). Vehicle Damaged Detection Using Deep Learning. International Research Journal of Modernization in Engineering Technology and Science, 4(09).

[12] Gayathri, P., Dhavileswarapu, A., Ibrahim, S., Paul, R., & Gupta, R. (2023). Exploring the potential of VGG-16 architecture for accurate brain tumor detection using deep learning. Journal of Current Medical Research and Opinion, 2(2), Article 23056. https://doi.org/10.57159/gadl.jcmm.2.2.23056

[13] A. Ajit, K. Acharya, and A. Samanta, "A Review of Convolutional Neural Networks," in 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), pp. 1–5, IEEE, feb 2020.

[14] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception architecture for computer vision. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2818-2826. https://doi.org/10.1109/CVPR.2016.308.

[15] Jing, Y., Li, C., Du, T., & Zhang, Y. (2023). A comprehensive survey of intestine histopathological image analysis using machine vision approaches. Computers in Biology and Medicine. Advance online publication. https://doi.org/10.1016/j.compbiomed.2023.106810.

[16] Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946. Retrieved from https://arxiv.org/pdf/1905.11946v5.

[17] DeepKapha. (2020, August 25). Implementing DenseNet-121 in PyTorch: A step-by-step guide. Medium. https://medium.com/deepkapha-notes/implementing-densenet-121-in-pytorch-a-step-by-step-guide-c0c2625c2a60