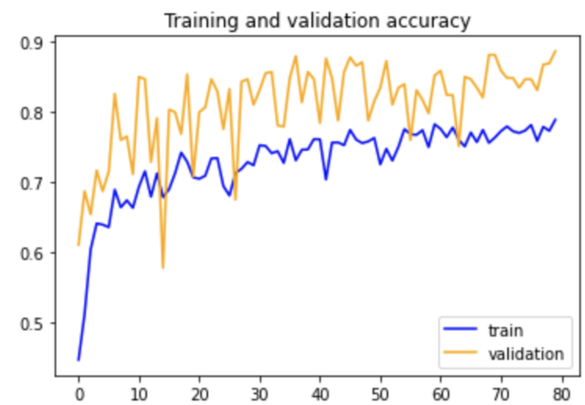
Sherry Zhao (txz270)
ECSE484 Final project report

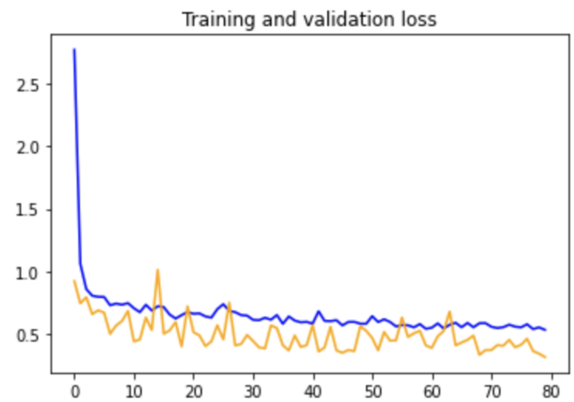
Settings

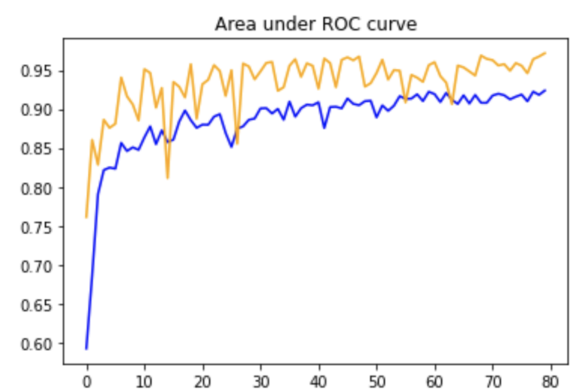
For the experiments in this project, I use the Multiclass-image-dataset-airplane-car-ship dataset. I use 5000 images for training and 582 images for validation. The batch size for both training and validation is 32. I use the image data augmentation to increase the variety of the images. I use Adam optimizer for neural networks except for the EfficientNet and the optimizer comparison experiments.

Compare variants of Residual Neural Network

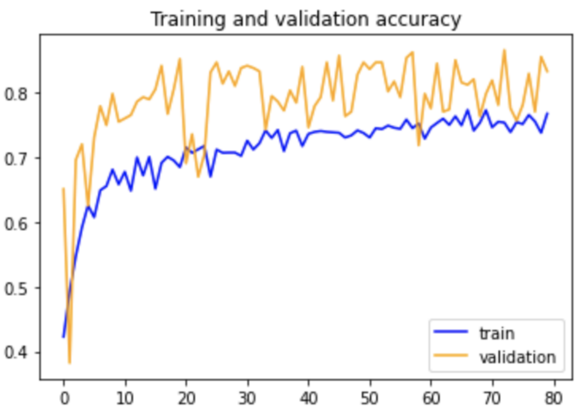
ResNet50

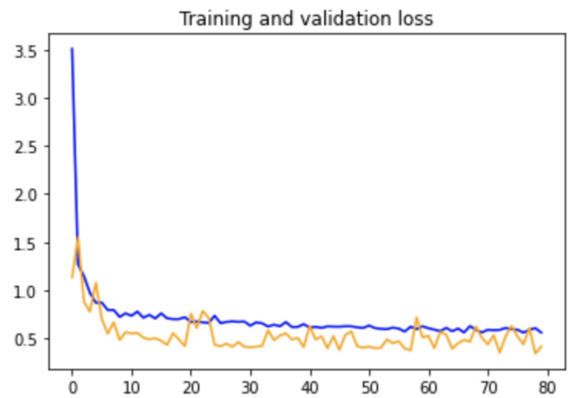


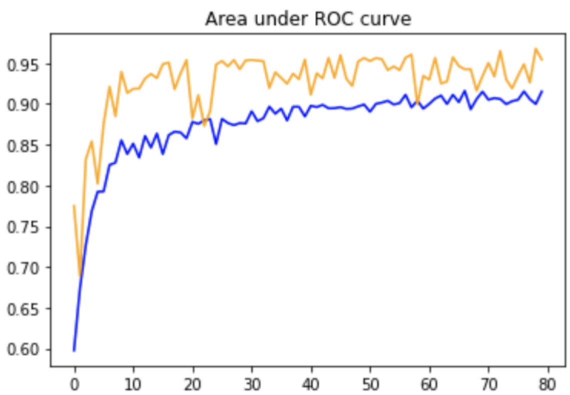




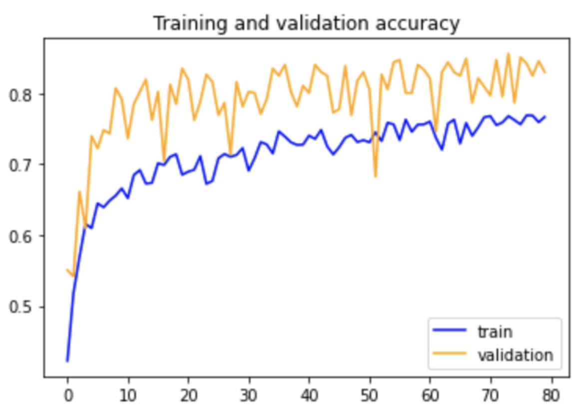
ResNet101

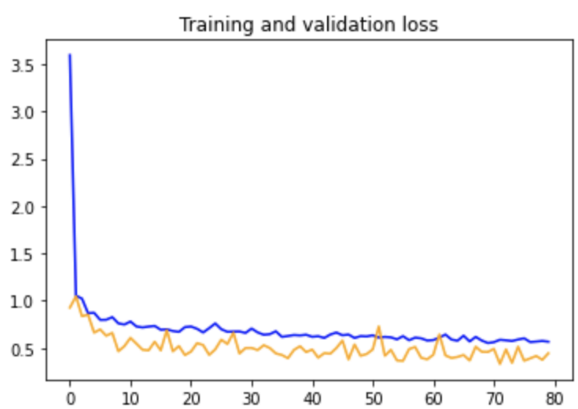

Training and validation accuracy
0.8
0.7
0.6
0.5
0.4
0 10 20 30 40 50 60 70 80
train
validation

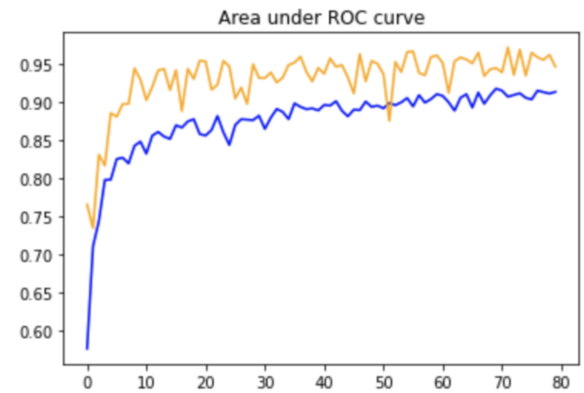

Training and validation loss
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0 10 20 30 40 50 60 70 80


Area under ROC curve
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0 10 20 30 40 50 60 70 80

ResNet152


Training and validation accuracy
0.8
0.7
0.6
0.5
0 10 20 30 40 50 60 70 80
train
validation


Training and validation loss
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0 10 20 30 40 50 60 70 80

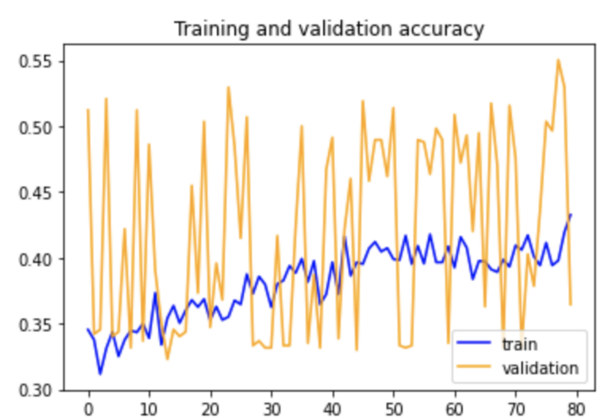


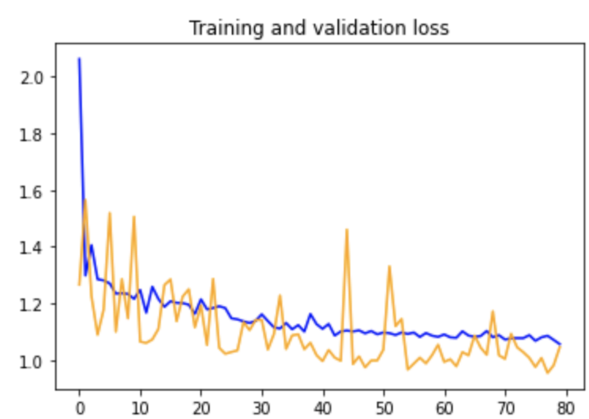
The accuracy, loss, and AUC (area under the ROC curve) of ResNet50, ResNet101, and ResNet152 do not have significant differences. The training accuracy of all three neural networks are converging to 0.8, and the validation accuracy of all three neural networks are converging to 0.9. The training loss of all three neural networks converges to 0.5, and the validation loss of all three neural networks converges to 0.3. The training AUC of all three neural networks converges to 0.93, and the validation AUC of all three neural networks converges to 0.97. However, the runtime of ResNet50 is less than ResNet101, and the runtime of ResNet101 is less than ResNet152. Because ResNet50 has the least number of layers, ResNet101 has more layers than ResNet50, and ResNet152 has the most layers, more calculations are needed for more layers, so the runtime increases. Because the performance of accuracy doesn't change much, it shows that ResNet50 has enough complexity in its layers for the classification.

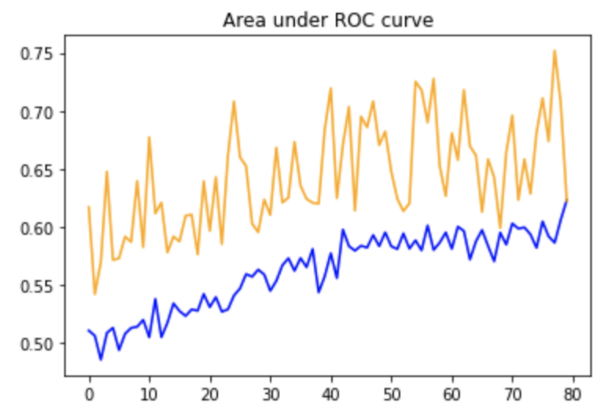
Compare variants of EfficientNet

Because the Adam optimizer works very poorly with the EfficientNet during experiments, I used SGD optimizer for EfficientNet. The rest of the settings are the same.

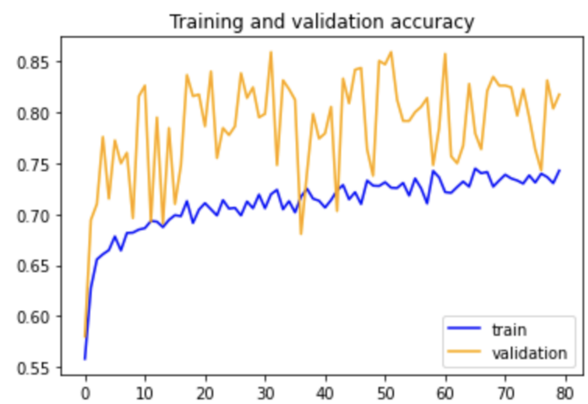
EfficientNetV2B0

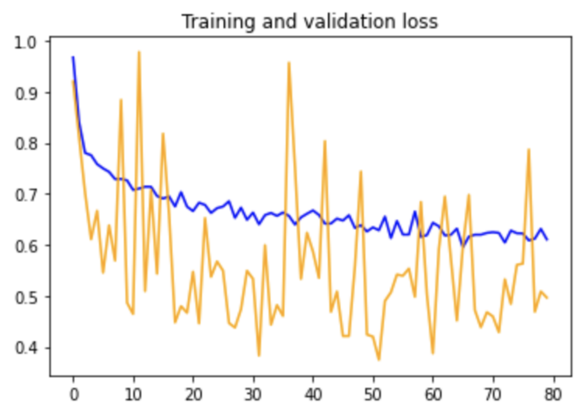


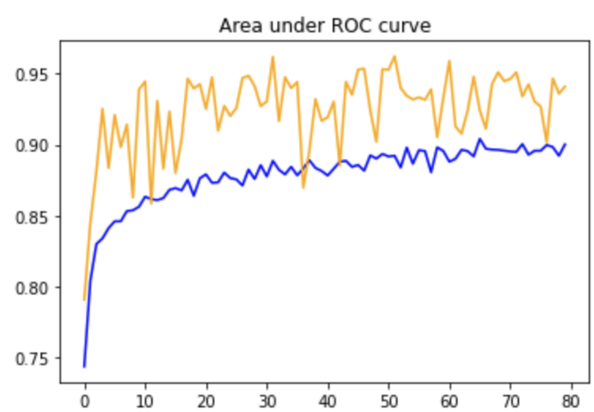




EfficientNetV2S



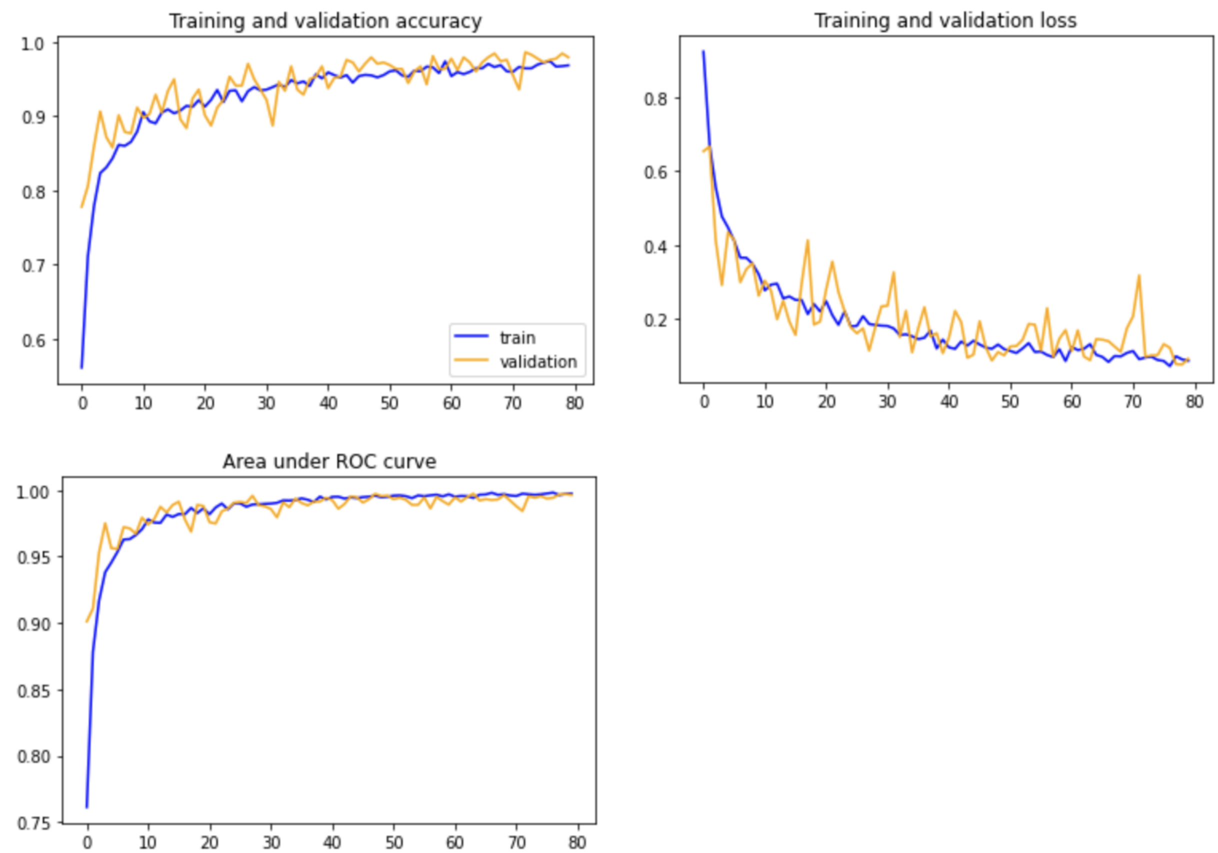




EfficientNetV2B0 performs very poorly for the dataset. Its low accuracy shows that it cannot capture the patterns to correctly classify the images. The training accuracy of EfficientNetV2S converges to 0.75, and training loss converges to 0.6, and the AUC converges to 0.9. EfficientNetV2S has more parameters than EfficientNetV2B0, which may be the reason for better performance. The performance of EfficientNetV2S is slightly lower than ResNets, and there is also an underfitting problem in EfficientNet and its validation performance is very unstable. According to Hoang and Jo (2021), EfficientNet has mobile inverted bottlenecks, which remove the channel connection to reduce computation, leading to reduction of complexity of the neural network, and the underfitting problem may be caused by the lack of complexity.

Compare ResNet50, EfficientNet, and Conv2D
Because ResNet50 has the best performance in ResNet and EfficientNetV2S has the best performance in EfficientNet, we use them to compare with the Convolutional neural network. The performance of ResNet and EfficientNetV2S results are shown in the previous sections.

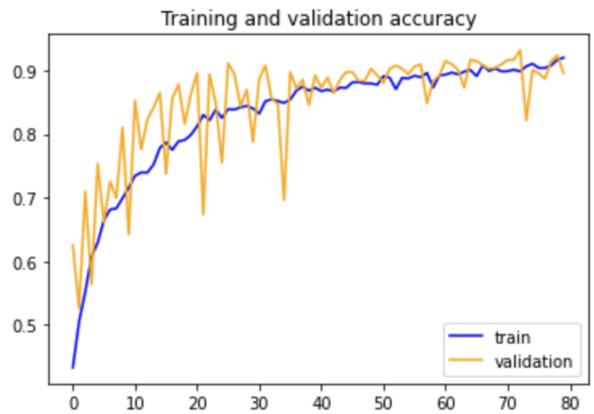
Conv2D

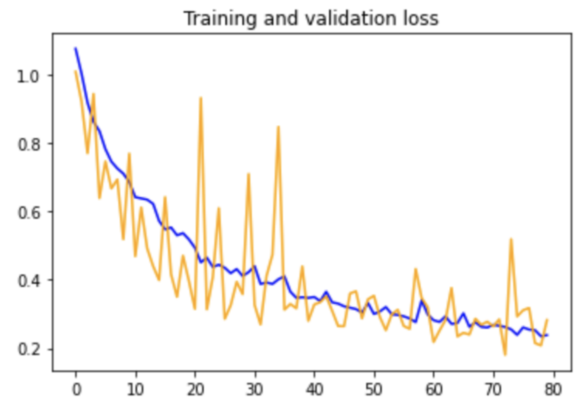


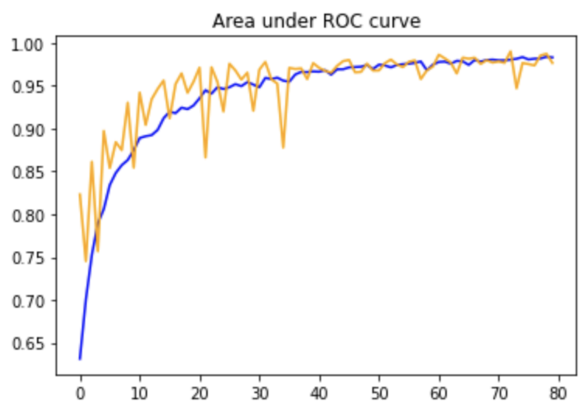
According to the results, the Conv2D has the best performance. Conv2D does not have the underfit problem, and it converges to higher accuracy, higher AUC, and lower loss than ResNet50 and EfficientNetV2S. I have some idea about why Conv2D performs better than the other two neural networks. I used transfer learning on ResNet and EfficientNet. The ResNet and EfficientNet use the weights from ImageNet and the layers in ResNet and EfficientNet are not trainable according to my setting. Only the two dense layer in the model can adjust according to the training images, while all the layers in the Conv2D model are trainable. The untrainable layers make the learning limited and have low performance, and also lead to the underfit problem. This may indicate that the weights from ImageNet are not suitable for my dataset. I should make the layers, or some of the layers, trainable in the ResNet and EfficientNet to get better performance.

Compare optimizers

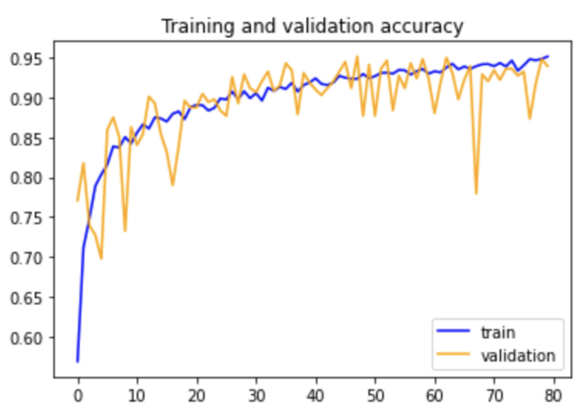
SGD

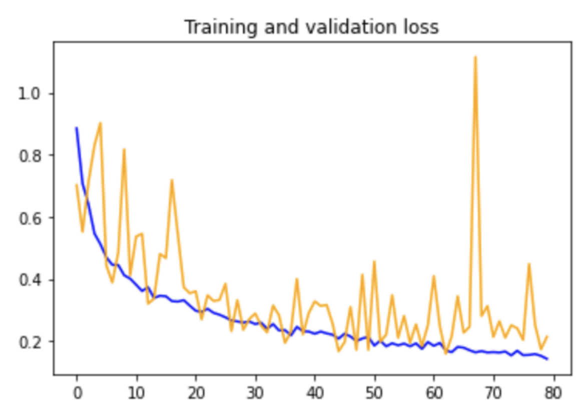


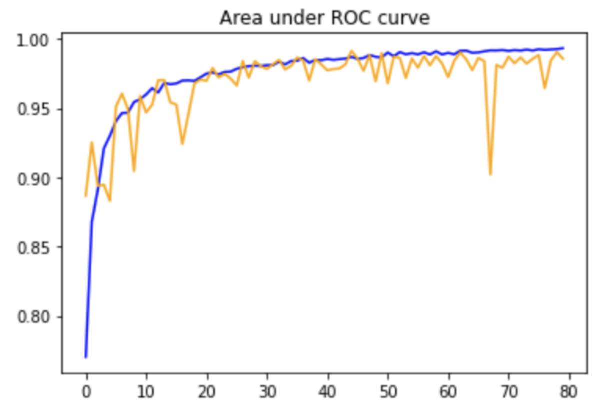




RMSprop







Comparing the optimizers, the Adam and RMSprop optimizer converge to similar accuracy, loss and AUC for both training and validation images, while SGD optimizer converges to a slightly lower accuracy, higher loss, and lower AUC. The performance of the training set for all three optimizers is stable. Adam has the most stable performance of validation set, RMSprop has less stable performance of validation set and includes a sudden low peak during the training, and SGD has the most unstable performance of validation set. According to Dozat (2016), Adam can perform better than SGD because Adam has adaptive step size to decrease the gradient noise. I use the default step size for Adam and SGD, and set the learning rate of RMSprop to 0.0001. It is possible that because Adam has adaptive step size, it adjusts the step size according to the gradient and performs the best. RMSprop is set to a small learning rate, so RMSprop performs better than SGD.

Conclusion

In conclusion, according to my experiments, the Conv2D with Adam optimizer has the best result for classifying the Multiclass-image-dataset-airplane-car-ship dataset. The low performance of ResNets and EfficientNets may be caused by the unsuitable architecture of the neural network or the untrainable weights from transfer learning. Adam is the best optimizer for Conv2D classification because of its adaptive step size during learning.

Reference:

[1] Dozat, Timothy. "Incorporating nesterov momentum into adam." 2016.
[2] Hoang, Van-Thanh, and Jo, Kang-Hyun. "Practical Analysis on Architecture of EfficientNet." 2021 14th International Conference on Human System Interaction (HSI). IEEE, 2021.