A Voting System of 3 Deep Learning Approaches for Pneumonia Detection in X-ray Images

Nissopoulou Theopisti Xeni
tnissopoulou@ihu.edu.gr

Prof. Kostantinos Diamantaras
k.diamantaras@ihu.edu.gr

Ntampakis Nikolaos
nntampakis@ihu.edu.gr

***ABSTRACT* – Pneumonia is one of the major reasons of deaths caused by lungs diseases around the world. It is known that during 2021 pneumonia caused more than 2.5 million deaths [1] Especially during Covid-19 period, this thread is getting bigger. It is always a challenging task to detect pneumonia in x-rays often due to limited professional radiologists in hospitals. In this paper, we develop a voting system of 3 models to successfully distinguish between no infection, bacterial and viral pneumonia in chest x-ray images. The proposed model's performance (accuracy: 88.65%) was evaluated in the 40% of the given dataset. The original Pneumonia Classification Dataset was shared through Kaggle, and the data are mostly taken from the dataset Chest X-Ray Images (Pneumonia) [2]**
Keywords: Image Classification, Pneumonia, X-rays

I. Data & Problem Description

The given dataset [3] was mostly taken from the dataset Chest X-Ray Images (Pneumonia) [2] and was organized into 2 folders (train, test). The train folder contained 4672 images (JPG) of 3 classes (1227 normal x-ray images, 2238 with bacterial pneumonia and 1207 images of viral pneumonia). Also, there was a csv file with the corresponding labels of the train images (class_id:0 for normal, class_id:1 for bacterial and class_id:2 for viral).

Figure 1. Sample of train/test images

Finally, there was a test folder of 1168 images on which our model was evaluated. Based on the description of the dataset, the images were selected from pediatric patients of one to five years old.

The final goal is to correctly classify the x-rays in the correct class and to achieve a performance comparable to human experts. The proposed model may aid in early diagnosis and treatment, resulting in improved clinical outcomes.

II. Models' Description

The proposed process consists of 3 models and a final voting system which decides the class based on the most frequent prediction across the previously mentioned models. The models that participated on that process were a Vision Transformer model (ViT), the ConvNet for the 2020s model (ConvNeXt) and a network which is a concatenation of other 3 networks (Xception, ResNet50V2 and ResNet152V2 networks).

A. ViT

Inspired by NLP Transformers, in 2021 images Transformers were introduced [4]. The images are split into patches and provide the sequence of linear embeddings of those as an input to a Transformer. The model is trained on image classification in a supervised manner. Generally, Vision Transformers seem to outperform in large datasets when the transfer knowledge (pre-trained) seems to also have great results in smaller datasets.

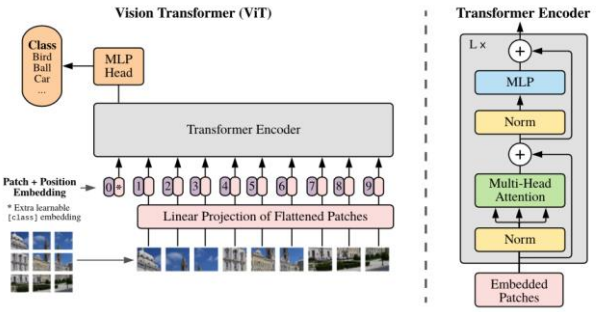

Figure 2. ViT overview

The implementation of the model (as shown in Figure 2) followed as close as possible the original one, used for NLP architectures.

The Transformer receives an input of image in 2D. and reshapes it into a flattened 2D patches ($xp \in R\ N\times(P\ 2\ \cdot C)$) where the resolution is (H,W) and the number of image's channels C. Also (P,P) is the resolution of each image patch and N = HW/P^2 the number of patches. The outpour of this procedure called "patch embeddings". There is a classification head also, implemented by an MLP with one hidden layer.

To retain positional information, the position embeddings are added to the "patch embeddings". The position embeddings are in the form of 1D standard learnable. The encoder (alternating layers of self-attention and MLP with a GELU) is applied before and after every block. The ViT seem to also have much less image-specific bias than CNNs due to locality of neighbour's

structure of CNNs. In ViT the self-attentional layers are global.

The initial model was trained in ILSVRC-2012 ImageNet dataset with 1k classes and 1.3M images and some supersets of it [5]. The gained knowledge was transferred to various benchmark tasks like CIFAR-10/100 [6], Oxford-IIIT pets [7] etc. Also, regarding the model's variants, a summary could be found in Table 1.

Model	*Layers*	*Hidden size D*	*MLP size*	*Heads*	*Params*
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Table 1. ViT variants

In the table 1 "Heads" means the input patches size. Also, for the baseline CNNs, the experiment used a ResNet with group normalization [8]. Based on the results the Vision Transformers outperformed ResNet baseline on all datasets under consideration. (Table 2)

Dataset	*ViT-H14*	*ResNet152x4*
ImageNet	**88.55**	87.54
ImageNet ReaL	**90.72**	90.54
CIFAR-10	**99.50**	99.37
CIFAR-100	**94.55**	93.51
Oxford-IIT Pets	**97.56**	96.62
Oxford Flowers-102	**99.68**	99.63
VTAB (19 tasks)	**77.63**	76.29

Table 2. Comparison in terms of accuracy in benchmarks

B. ConvNeXt

ConvNeXt is a construction which components are ConvNets modules, and it competes favorably with ViT in terms of accuracy [9] and outperforming Swin Transformers in COCO dataset, maintaining at the same time the simplicity of ConvNets. (Figure 3)

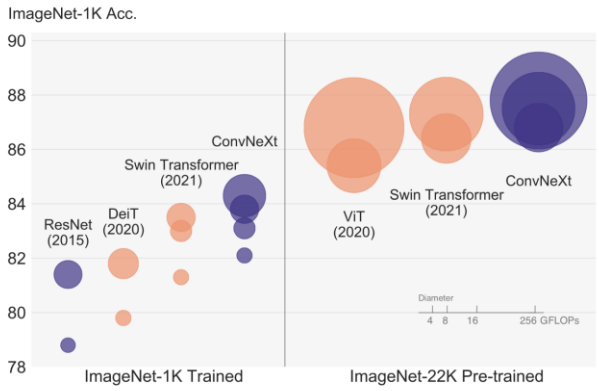

Figure 3. ConvNeXt performance

The roadmap of a ConvNext follows a series of steps and tries to investigate the designs of Swin Transformers [10] when keeping the simplicity of a ResNet-50. The baseline was also the original ResNet-50. In Figure 4 we can see the series of steps starting from a ResNet-50 trying to train it with similar techniques as ViT and then a list of designing decisions which analyzed below:

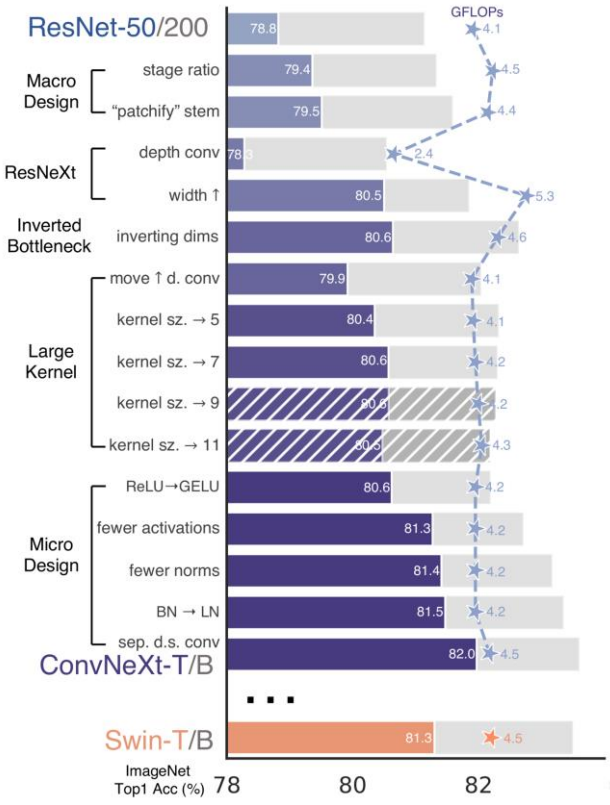

Figure 4. ConvNeXt roadmap

Macro Design in this step the macro network design of a Swin Transformer is analyzed. The output is to change the number of blocks in each stage in ResNet to (3,3,9,3) improving the accuracy to 79.4%. Also, the ResNet-style stem cell is replaced with a patchify layer (4x4 stride) improving the accuracy from 79.4% to 79.5%.

ResNeXt is a step in which more groups are implemented, expanding the width. As a result of more groups both FLOps and accuracy were reduced significantly. Then, the network width is increased in the same size as Swin-T's and the performance is increased also (accuracy 80.5%)

Inverted Bottleneck is a hidden dimension of an MLP block, and it is 4 times wider than the input dimension. In this step the inverted bottleneck design is implemented as shown in Figure 5(b) and the accuracy slightly improved (80.6%)

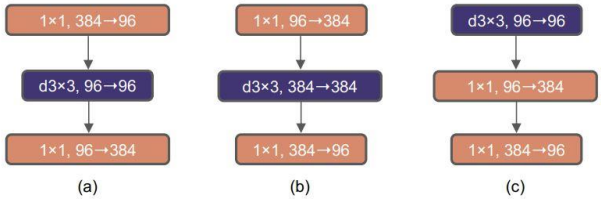

Figure 5. (a) ResNeXt block, (b) Inverted Bottleneck block, (c) Moved position of spatial depthwise conv layer

Large Kernel Sizes is a step in which large kernel-sized convolutions for ConvNets are used. As a first step here the depthwise conv layer is moved up Figure 5 (c). reducing temporarily the performance. Then the kernel size in increased from 3x3 to 7x7 after various kernel wise experiments increasing again the accuracy to 80.6%.

Micro-design is mostly focused on the use of different activation functions and normalization layers. Firstly, the

ReLu activation function is replaced with GELU [11] which is a smoothest variant of ReLu and the accuracy remained the same. Then proceeded with fewer activation function (single GELU per block) and fewer normalization layers boosting the performance to 81.4% as shown in Figure 6.

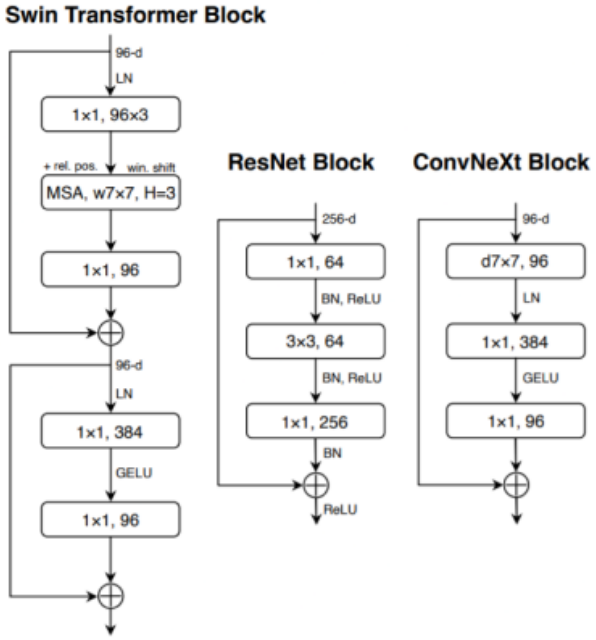

Figure 6. Block designs for ST, RN and CNX

Also, a Layer Normalization was applied instead of Batch Normalization. Finally, a separation of downsampling layers is performed leading to an accuracy of 82%, and this concludes the ConvNeXt roadmap.

ConvNeXt outperformed both RegNet and EfficienNet and some ViT Transformers variants (DeiT and Swin) in ImageNet-1K and ImageNet-22K datasets (Table 3)

Model	*ImageNet-1K*	*ImageNet-22K*
RegNet	82.9	85.4
EffNetV2	85.7	87.3
DeiT-B	81.8	---
Swin-B	84.5	---
Swin-L	---	87.3
ViT-L/16s	---	86.8
ConvNeXt-L	**85.5**	87.5
ConvNeXt-XL	---	**87.8**

Table 3. Comparison in terms of accuracy of various models for ConvNeXt

C. Concatenated Model

This custom model was created by utilizing multiple features extracted by three robust networks (ResNet152V2, ResNet50V2 and Xception).

As we can see in Figure 7, the input will be, of course, the training images of 3 classes. Then we take the same output – in terms of dimensions - of each network with transferred knowledge (weights will come from imagenet) as 10*10*2048 vectors. Finally, we concatenate them and pass them thought a fully connected layer using a dropout (50%) and a softmax for the final classification.

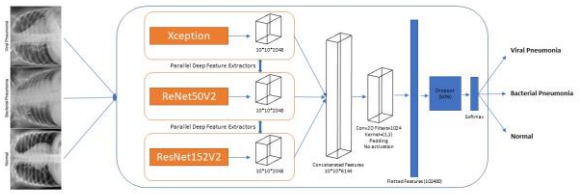

Figure 7. Concated model architecture

Xception model[12] is an interpretation of Inception modules in CNN and it is based entirely on depthwise separable convolution layers (essentially it is a linear stack of separable layers with residuals connections). As sown in Figure 8 the Xception architecture has 36 convolutional layers. As the model was used for image classification, a logistic regression layer follows.

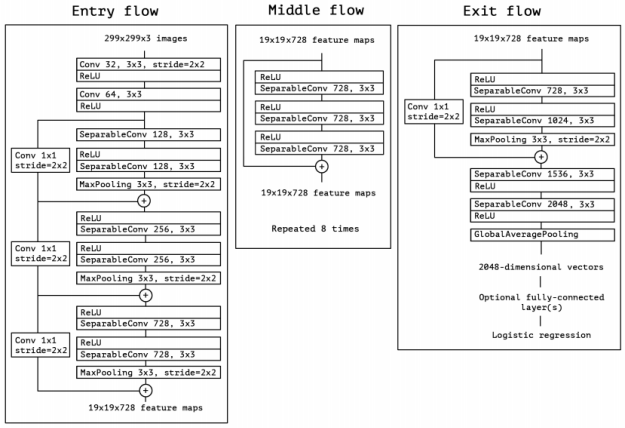

Figure 8. Xception architecture

The Xception model outperformed on ImageNet classification to VGG-16, ResNet-152 and Inception V3 as shown on Table 4.

Model	*Top-1 Acc*	*Top-5 Acc*
VGG-16	0.715	0.901
ResNet-152	0.770	0.933
Inception V3	0.782	0.941
Xception	**0.790**	**0.945**

Table 4. Comparison in terms of accuracy of various models for Xception

ResNet50V2 and ResNet 152V2 [13] are modified versions of the original ResNet50 (Figure 9) and ResNet152 (Figure 10) models achieving better results. Both are convolutional neural network that were trained on more than a million images from the ImageNet. Both initial and version2 ResNet introduce the concept of residual networks in order to solve the problem of the vanishing/exploding gradient. The used technique (called skip connections) skipping some intermediate layers and connects directly to the output, so instead of layers learn the underlying mapping, we allow network to fit the residual mapping.

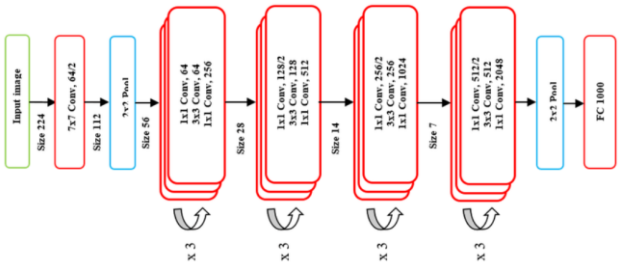

Figure 9. ResNet50 architecture

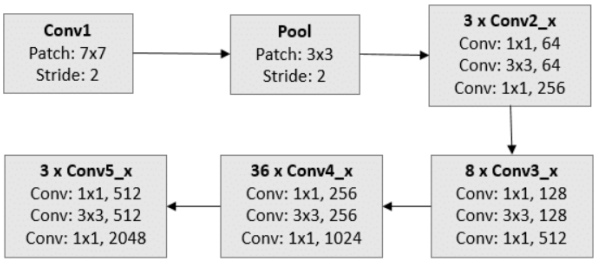

Figure 10. ResNet152 architecture

The basic difference between version 1 and 2 of ResNet50 is all about using the pre-activation of weight layers instead of post-activation [14]. Figure 11 shows the architecture of post-activation (v1) vs pre-activation (v2) of ResNet.

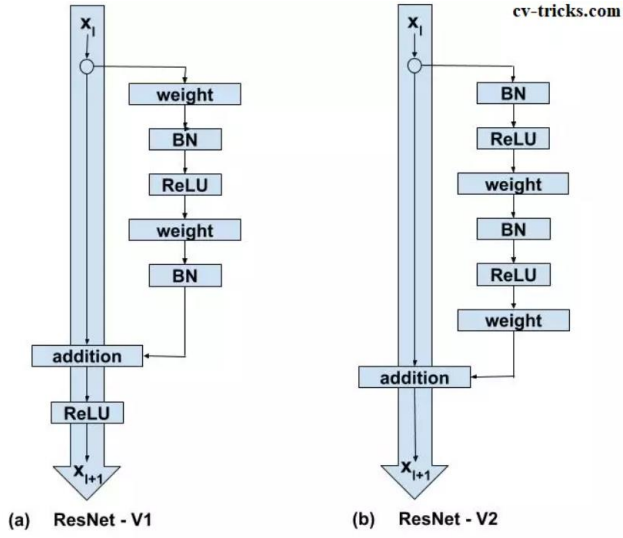

Figure 11. Basic differences between versions of ResNet

Concatenated neural network [15] was designed to concatenate all the extracted features of the previous 3 models to a convolutional layer. The last layer is connected to the classifier. To extract a more valuable semantic feature, a convolutional layer with a kernel of 1x1 and 1024 features was added. Finally, a dropout technique was used to prevent the model from overfitting. The use of a softmax lead us to the final classes.

D. Voting System

The proposed voting system is a majority decision among 3 models. A common file was created with the predictions of all 3 models per image in test set, picking the most frequent decision among 3. In cases of a triple disagreement, the predicted class of the best model was used.

III. EXPERIMENTS & RESULTS

The approach we followed was common across all 3 of our models. We made an initial experiment; we fine-tuned the models and finally we had an experiment with multiple epochs trying to pick up the best version of our model (specific epoch) in terms of validation accuracy. Also, we tried our best to avoid overfitting, keeping an eye both in the train accuracy and validation loss. In all 3 models we used a train/validation separation of 80/20 and data augmentation.

Regarding ViT, a reshape of the images in 144*144 was needed along with the (hyper)parameters of Table 4.

(Hyper)parameters	
Learning rate	0.001
Weight decay	0.0001
Batch size	256
Patch size	6
Transformers layers	8
Numbers of heads	4
MLP head units	[2048,1024]
Ooptimizer	AdamW
Epochs	100

Table 4. (Hyper)parameters of ViT

These (hyper)parameters led us to 576 patches/image and 108 elements/patch (Figure 12).

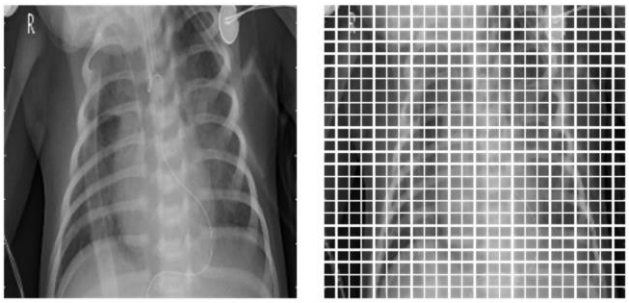

Figure 12. Initial image vs Image with segmented patches

Using a dropout rate of 50%, and a softmax function in the final layer we get a validation accuracy of **81.6%** for the model produced by epoch 66 without signs of overfitting (Figures 13 and 14)

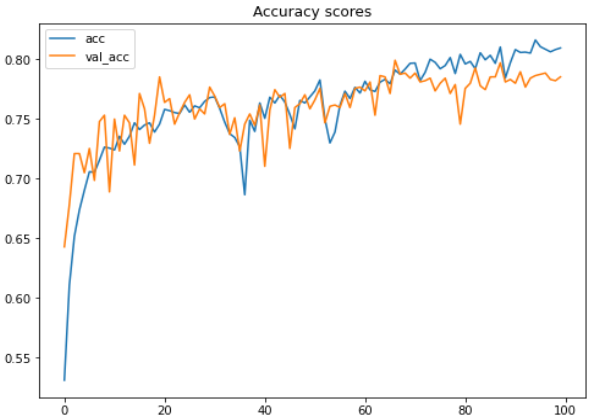

Figure 13. Train and Validation accuracy (ViT)

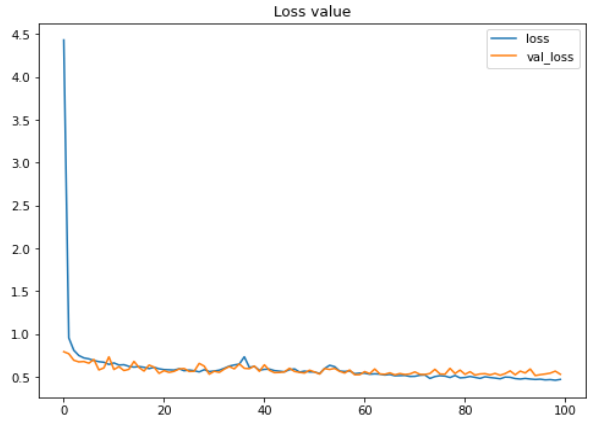

Figure 14. Train and Validation loss (ViT)

In terms of our ConvNeXt model we used the large-224 version of it. The model came as pretrained from https://huggingface.co/. Again, here we used the following (hyper)parameters (Table 5)

(Hyper)parameters	
Learning rate	5e-5
Batch size	8
Optimizer	AdamW
Epochs	10

Table 5. (Hyper)parameters of ConvNeXt

Our second model achieved a validation accuracy of **86.3%** for the model produced by epoch 10 without very strong signs of overfitting (Figures 15 and 16)

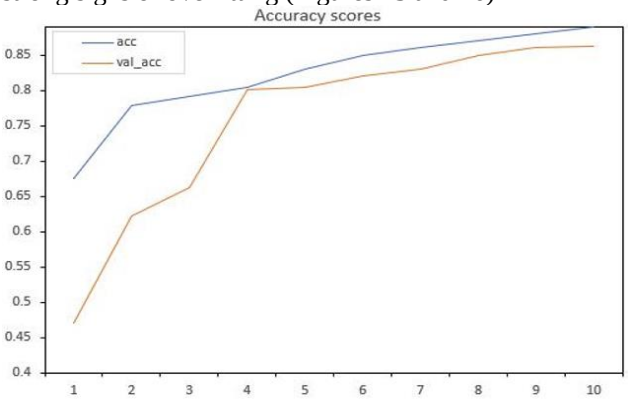

Figure 15. Train and Validation accuracy (ConvNeXt)

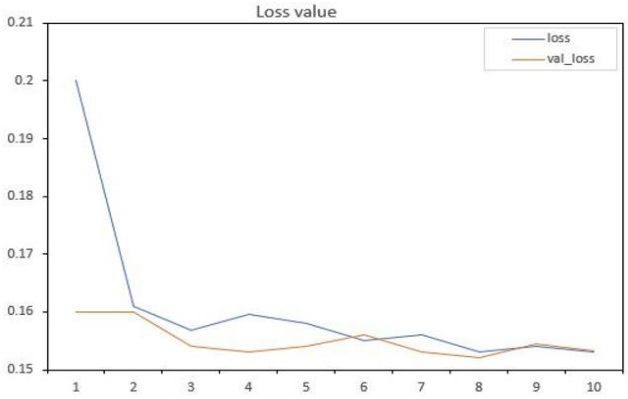

Figure 16. Train and Validation loss (ConvNeXt)

Our last model, which is a concatenation of 3 models was based on pretrained weights of imagenet dataset. We used the (hyper)parameters of Table 6.

(Hyper)parameters	
Learning rate	0.0001
Batch size	8
Optimizer	Nadam
Epochs	35

Table 5. (Hyper)parameters of Concatenated Model

Using a dropout rate of 50%, and a softmax function in the final layer we get a validation accuracy of **85.4%** for the model produced by epoch 24 with indications of overfitting (Figures 17 and 18)

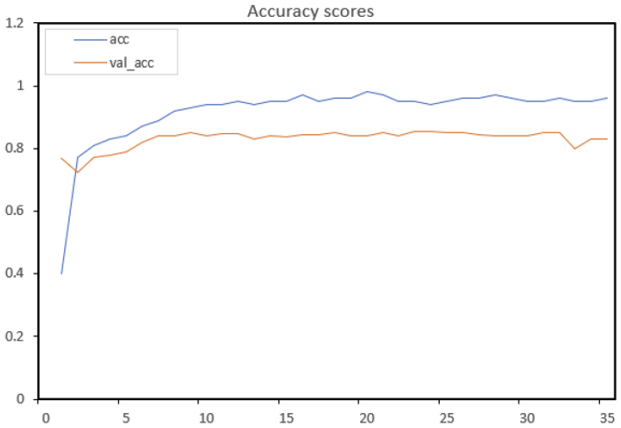

Figure 17. Train and Validation accuracy (Concatenated Model)

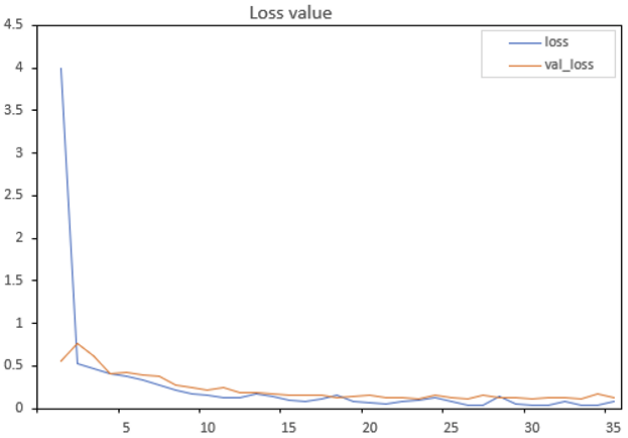

Figure 18. Train and Validation loss (Concatenated Model)

Finally, our voting system took place by picking the major predicted class per image among the 3 models. A triple disagreement was observed only in 5 cases out of 1168. In those cases, the class predicted by the best model (ConvNeXt) was picked achieving an accuracy of **88.7%**. Summarized results could be found in Table 6.

Model	*Accuracy*
ViT	81.6
ConvNeXt	86.3
Concatenated Model	85.4
Voting System	**88.7**

Table 6. Accuracy per model

IV. CONCLUSION

In this paper, we presented a voting system of 3 models (ViT, ConvNeXt and a concatenated neural network based on Xception, ResNet50V2 and ResNet152V2 networks) for classifying chest X-ray images into three classes of normal, viral and bacterial pneumonia. The datasets were provided via Kaggle platform (www.kaggle.com/competitions/detect-pneumonia-spring-2022) which consist of 4672 train images (1227 of class 0 or normal, 2238 of class 1 or bacterial pneumonia and 1207 of class 3 or viral pneumonia) and 1168 test images. We managed to achieve a total accuracy of 88.7% in the test set. We hope that our proposed method will be useful for medical diagnosis. Future larger datasets of X-ray images and the use of attention-based models on top of our proposition could successfully increase the accuracy of classification.

CITATIONS

[1] The Hospital Clínic de Barcelona 2021, Interview with Dr Catia Cilloniz, accessed 14 May 2022, <https://www.clinicbarcelona.org/en/news/pneumonia-causes-2-5-million-deaths-around-the-world-each-year>

[2] Kermany, Daniel; Zhang, Kang; Goldbaum, Michael (2018), "Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification", Mendeley Data, V2, doi: 10.17632/rscbjbr9sj.2

[3] Kaggle 2018, Paul Mooney, Chest X-Ray Images (Pneumonia), accessed 14 May 2022, <https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia>

[4] Alexey Dosovitskiy∗,† , Lucas Beyer∗ , Alexander Kolesnikov∗ , Dirk Weissenborn∗ , Xiaohua Zhai∗ , Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby∗,† ∗ equal technical contribution, † equal advising Google Research, Brain Team, AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE, Published as a conference paper at ICLR 2021 <https://arxiv.org/pdf/2010.11929v2.pdf>

[5] J. Deng, W. Dong, R. Socher, L. -J. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848, <https://ieeexplore.ieee.org/document/5206848>

[6] Alex Krizhevsky, "Learning Multiple Layers of Features from Tiny Images", 2009, <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>

[7] O. M. Parkhi, A. Vedaldi, A. Zisserman and C. V. Jawahar, "Cats and dogs," 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3498-3505, doi: 10.1109/CVPR.2012.6248092. <https://ieeexplore.ieee.org/document/6248092>

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep Residual Learning for Image Recognition", 2015, <https://arxiv.org/abs/1512.03385>

[9] Zhuang Liu1,2* Hanzi Mao1 Chao-Yuan Wu1 Christoph Feichtenhofer1 Trevor Darrell2 Saining Xie1† 1Facebook AI Research (FAIR) 2UC Berkeley,"A ConvNet for the 2020s", 2022 <https://arxiv.org/pdf/2201.03545v2.pdf>

[10] Ze Liu†* Yutong Lin†* Yue Cao* Han Hu*‡ Yixuan Wei† Zheng Zhang Stephen Lin Baining Guo, Microsoft Research Asia, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows ", 2021, <https://arxiv.org/pdf/2103.14030.pdf>

[11] Dan Hendrycks, Kevin Gimpel, "Gaussian Error Linear Units (GELUs)", version v4, 2020 <https://arxiv.org/abs/1606.08415>

[12] Francois Chollet Google, Inc., "Xception: Deep Learning with Depthwise Separable Convolutions", 2017, <https://arxiv.org/pdf/1610.02357.pdf>

[13] He, K., Zhang, X., Ren, S., Sun, J. (2016). Identity Mappings in Deep Residual Networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science(), vol 9908. Springer, Cham. https://doi.org/10.1007/978-3-319-46493-0_38<https://link.springer.com/chapter/10.1007/978-3-319-46493-38?utm_source=getftr&utm_medium=getftr&utm_campaign=getftr_pilot>

[14] CV-Tricks.com, Learn Machine Learning, AI & Computer vision, "Detailed Guide to Understand and Implement ResNets" by ANKIT SACHANL accessed 14 May 2022 <https://cv-tricks.com/keras/understand-implement-resnets/>

[15] Mohammad Rahimzadeh, Abolfazl Attar, A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X ray images based on the concatenation of Xception and ResNet50V2, Informatics in Medicine Unlocked, Volume 19, 2020, 100360, ISSN 2352-9148, https://doi.org/10.1016/j.imu.2020.100360, https://www.sciencedirect.com/science/article/pii/S2352914820302537>