Capstone project Proposal

Domain Background

The project I have chosen for my capstone project is images classification of X-ray images of people lunges some consist of 2 categories some are: (Normal and Pneumonia) - *"Pneumonia is an infection in airways or lungs usually caused by infection with viruses or bacteria or less commonly by other microorganisms"* [1] .

The reason for I think it will be a relevant topic is because I think AI will be the future in for Healthcare and nursing and will be the next step in the future to get into more advanced tech to help other people, some have the potential to save and help a lot beyond what we use today. Therefore, have I chosen x-ray image classification with neural networks and also because neural networks are a subject I intend to work with in the future so it will be the most logical I also use it in my capstone project.

One Related topic some are like my project in many ways is Deepcare some stands for a deep, dynamic neural network(https://truyentran.github.io/papers/pakdd_16.pdf). Deepcare is an "end-to-end deep dynamic neural network that reads medical records, stores previous illness history, infers current illness states and predicts future medical outcomes" cited from page 1/12 in the research paper [2]. It does this out from (ERM) electronic medical records some contains "details including family history, reason for initial complaint, diagnosis and treatment, prescription medications, lab tests, and other vital details needed to provide assistance to each patient" [3] out from these record will Deepcare have the ability to predict illness in the future of patients some are similar to my project where I just diagnose the patients out from X-rays images instead ERM's to determine if there have a illness.

problem statement

The problem some can be help with machine learning, in this case, is to diagnose the partitions some have Pneumonia in their lungs out for X-ray images of the patients.

The use for such an algorithm can be to help out doctors with determining if a patient has Pneumonia or not based on finding patterns in the X-ray images some might be a challenging thing to find for a human eye in some cases and base on that can help out patients to get a correct diagnosis.

Datasets and Inputs

The data some I will be using is from Kaggle.com [4] it is images there are "*selected from retrospective cohorts of pediatric patients of one to five years old from Guangzhou Women and Children's Medical Center, Guangzhou. All chest X-ray imaging was performed as part of patients' routine clinical care*." [4]

"The dataset is organized into 3 folders (train, test, val) and contains subfolders for each image category (Pneumonia/Normal). There are 5,863 X-Ray images (JPEG) and 2 categories (Pneumonia/Normal)" [4] and in each category are there:

- Train 5,219 Files, Normal 1,342 Files, Pneumonia 3,876 Files, .DS_Store 3 File
- Val 19 Files, Normal 9 Files, Pneumonia 9 Files, .DS_Store 3 File
- Test 625 Files Normal 234 Files, Pneumonia 390 Files, .DS_Store 1 File

all the images are 1 color dimensions (black and white). The images have all different dimensions so there is not a default resolution for the images in the dataset. [4]

The original data is from data.mendeley.com and the Contributor(s) is: Daniel Kermany, Kang Zhang, Michael Goldbaum. [5]

Solution Statement

The problem can be solved by using a Convolutional Neural Networks(CNN) to perform images classification with training a model on the 2 categories(Normal and Pneumonia) out from the data there is provided from Kaggle [4] so the neural network will be able to classify other X-ray images and tell if a person will have Pneumonia in their lungs or if there are normal healthy lungs out form patterns in the images and thereby can help the healthcare system with diagnosing their patients with Pneumonia.

Benchmark Model

The model I intended to use will consist of 2x3 CNN layers with batch normalization in between every 2 CNN layers and max poling layers after every 2 CNN layers. I will then use one flatten layer so the tensor will be converted to 1D tensor and then will I use 2 dense layers with dropout and batch normalization.

The reason for I will be using dropout, batch normalization and max-pooling layers in my neural network are:

- Dropout layers will help the model with not overfitting
- Batch normalization makes it easy for the optimizer to work properly and thereby bring down the loss.
- Max-pooling for bringing down the data load so I can run it on my pc
- Flatten is for covering the data to a 1D tensor so the dense layer will accept it when feeding it from the CNN layers.

I will use the Keras library to make the model with and the data preprocessing. [6] The model some I intent to use in its vanilla form will look like this:

Conv2D input	Filters 32, input shape (64 ,64 ,1)
Batch Normalization	
Conv2D	Filters 32
MaxPooling2D	pool size (2, 2)
Conv2D	Filters 64
Batch Normalization	
Conv2D	Filters 64
MaxPooling2D	pool size (2, 2)
Conv2D	Filters 128
Batch Normalization	
Conv2D	Filters 128
MaxPooling2D	pool size (2, 2)
Flatten	
Dense	units 1,024
Batch Normalization	
Dropout	Rate (0.5)
Dense	units 512
Batch Normalization	
Dropout	Rate (0.5)
Dense output	Units 2

The reason I have chosen this structure is that it is a known tested way to structure a model some often give good results like the VGG16 model structure. [7]

Evaluation Metrics

I will fine tune the model with Keras evaluate generator [8] and there after use Keras F-beta score to compute the result as an evaluation metrics [9].

Project Design

For this dataset will I first downscale the training data so normal and Pneumonia have the same number of images. I will do this by removing the necessary amount from Pneumonia so the data will have 50/50 of images in each category.

I will then use data augmentation on the dataset to upscale the data some will account for the loss of data I have been removing.

I will then use the data generators form Keras API with flow directory [8] the reason for that is so I can load the data in batches when training the model instead of loading anything in simultaneously. The advances to that are so I can train the model with bigger datasets without overloading my pc. I will downscale the resolution on the images to 64x64 so there are small but still have some details. If the 64x64 show to be to small will I go for 128x128 instead.

Then I will compile the model with “Adam” optimizer [10]. The reason for using “Adam” is because it is a versatile optimizer some are lightweight and performs well on most dataset so I think it will be a good choice for this dataset as well.

After compiling the data will I train the model with the model I described under “Benchmark Model” section, for 10 – 50 epochs depending on how well the model performs on the data. I will be using ModelCheckpoint function [11] to save all the weights so the model can be reused in other models also in the in the future.

Lastly, will I use the Keras evaluate function [8] to fine tune, the model with and then compute the F-beta score some I described in “Benchmark Model” section to get the final measurement for the result of the model performers.

References

[1] »wikipedida Pneumonia,« [Online]. Available: https://en.wikipedia.org/wiki/Pneumonia.

[2] »Deepcare research paper,« [Online]. Available: https://truyentran.github.io/papers/pakdd_16.pdf.

[3] A. d. s. c. -. S. O'Connor, »How Do EHR Systems Work,« [Online]. Available: https://www.adsc.com/blog/how-do-ehr-systems-work.

[4] P. Mooney, »kaggle,« [Online]. Available: https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia.

[5] K. Z. M. G. Daniel Kermany, »data.mendeley,« [Online]. Available: https://data.mendeley.com/datasets/rscbjbr9sj/2.

[6] »keras documentation,« [Online]. Available: https://keras.io/layers/convolutional/.

[7] M. u. Hassan, »VGG16 – Convolutional Network for Classification and Detection,« [Online]. Available: https://neurohive.io/en/popular-networks/vgg16/.

[8] »Keras Model class API,« [Online]. Available: https://keras.io/models/model/.

[9] »faroit - Usage of metrics,« [Online]. Available: http://faroit.com/keras-docs/1.2.1/metrics/.

[10] J. Brownlee, »Gentle Introduction to the Adam Optimization Algorithm for Deep Learning,« [Online]. Available: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/.

[11] »Keras - Usage of callbacks,« [Online]. Available: https://keras.io/callbacks/.