Zoidberg2.0_2020_15

AMMAR Sana

CRISTIANINI Marceau

HEADLEY Ryan

HUCK Geoffroy

MASSON Antoine

EPITECH

Abstract

This project was done to create an image classifier with the ability to take chest x-rays as inputs and accurately diagnose whether the lungs are infected with pneumonia or not. A dataset of 5865 images were provided from registered doctors to train, test and validate the classifier. Three classifiers were chosen, based on the most commonly used and accepted in the computer vision industry: k nearest neighbour (kNN), support vector machine (SVM) and convolution neural networks (CNN). After building and training these classifiers, the CNN was concluded to be the most effective model at diagnosing pneumonia with an accuracy of 96%

Keywords: Pneumonia, machine learning, chest x-ray, support vector machine, k nearest neighbour, convolution neural network

Introduction

With the development of artificial intelligence, computer vision has advanced to the point where its accuracy, in some situations, can be above the level of human vision. This technology opens the door to a wide range of applications, one of which is analyzing medical images.

At the end of 2019, a new virus, COVID-19, swept the globe infecting billions and killing millions. Its rate of spread was so effective that tracking and understanding it was almost an insurmountable task. Since the average year has millions of people succumbing to respiratory illnesses, understanding whether one is connected to the new virus is a difficult task. This study was conducted to create an algorithm with the ability to take any number of chest X-rays and determine whether each is infected with pneumonia and of which kind. This is one step towards the goal of tracking the infection rate of patients with only the input of images.

Background

Pneumonia is an infection that inflames the air sacs in one or both lungs. The air sacs may fill with fluid or pus (purulent matter), which causes a cough with or more mucus, fever, chills, and difficulty breathing. With the evolution of Covid-19, hospital departments are overwhelmed and are looking for a computer tool that can help doctors quickly and massively detect pneumonia from X-ray images.

In 2017, 2.56 million people died from pneumonia. Almost a third of these people were children under the age of 5. WHO has estimated that 45,000 of these premature deaths were due to domestic air pollution. With improved diagnostic efficiency, many of these deaths can be reduced.

With the evolution of Covid-19, hospital departments are overwhelmed and looking for a computer tool that can help doctors quickly and massively detect pneumonia from the X-ray images given.

The objective of this project is to build an optimized machine learning model so that it can help radiologists detect pneumonia early from chest x-rays, and save many lives.

Material

Dataset

The dataset used consists of a total of 5865 grayscale X-ray images (jpeg) of the chest, where 5216 are used for training, 624 for testing and 8 for validation. Each of the three sets are further separated into healthy and pneumonic infected lungs. All images were provided from the registered doctors included in the project.

Tools

Since this project is founded in artificial intelligence with the goal of being completed and shared as quickly as possible, a jupyter notebook was created. This option not only facilitates collaborative work, but it takes advantage of the most commonly used language in data science: python.

Alongside python, the numpy, panda, sklearn and tensorflow libraries were incorporated to build on top of pre-built machine learning algorithms.

Results

This project utilized three different classifiers, kNN, SVM and CNN, to find the most suitable function to predict pneumonia from the dataset. Each classifier was tested with a variety of different parameters, and the following results denote the most accurate model created.

Outcome kNN

This was the first classifier analyzed and the images of the dataset were transformed into two forms: raw images and histograms. Initially a cross validation score was run on the classifier to determine a k value of 7 to be the most optimal. After running the classifier with this value, an accuracy of 91.18% and 84.28% were reached for raw images and histograms, respectively.

Later, confusion matrices were calculated (*see appendix*) as well as the precision, recall, f1-score and support. As denoted by the accuracy, the raw images produced higher results than the histograms, which can be seen in *Figure 1 and 2*.

	precision	recall	f1-score	support
NORMAL	0.94	0.76	0.84	357
PNEUMONIA	0.92	0.98	0.95	947
accuracy			0.92	1304
macro avg	0.93	0.87	0.89	1304
weighted avg	0.92	0.92	0.92	1304

Figure 1: kNN metrics for raw images

	precision	recall	f1-score	support
NORMAL	0.70	0.73	0.72	357
PNEUMONIA	0.90	0.88	0.89	947
accuracy			0.84	1304
macro avg	0.80	0.81	0.80	1304
weighted avg	0.84	0.84	0.84	1304

Figure 2: kNN metrics for histograms

Outcome SVM

The second classifier applied was the support vector machine. Unlike kNN, this classifier requires a model to be built and trained to be able to make predictions. All images were resized and divided into batches. Each batch was separately trained for 1500 iterations, starting with a random weight which is later modified, along with the loss. A learning rate of 1e-7, a regularization of 2.5e4 and a batch size of 200 were applied and resulted in a model that is 80.8%

accurate on the training data and 83.1% on the validation data. The ROC curve was calculated and can be seen in *Figure 3* below.

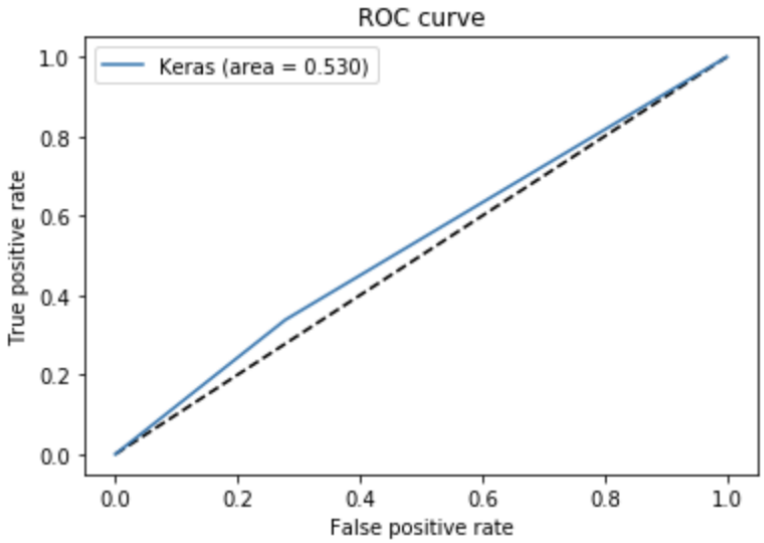


Figure 3: ROC curve for SVM analysis, with AUC labelled

The resulting data was not satisfactory to continue developing this classifier, and therefore all effort was applied to the CNN.

Outcome CNN

We find that the method CNN is pretty easy to use with an accuration very high. It's a very good algorithm to classify data. But the resources demanded are very high. It takes a lot of time and therefore you have to use it with different processes, such as freezing the model to be able to use it again to gain performance and learning time. It doesn't need a lot of epochs to be very accurate. But the more you iterate, the less false negatives you get.

The best thing about CNN is that you can use different sets of data during the training to make the model more effective. Here in *Figure 4*, you can see our different results with each epoch containing the train loss, valid loss and error rate that are important metrics for trusting the fiability of the model. As you see the more we train the less errors we get. But the time for each epoch is quite long. We also have the graph of loss depending on the learning rate, seen in *Figure 5*. Here we see that a learning rate of around 1e-4 drastically reduces the loss, and the best accuracy is around 1e-1, which is the max learning rate that we want.

epoch	train_loss	valid_loss	error_rate	accuracy	time
0	0.179052	0.110573	0.037575	0.962425	07:51
1	0.118281	0.102691	0.034159	0.965841	07:46
2	0.077100	0.084856	0.027327	0.972673	07:50

Figure 4: Accuracy, loss and other metrics for first 3 epochs

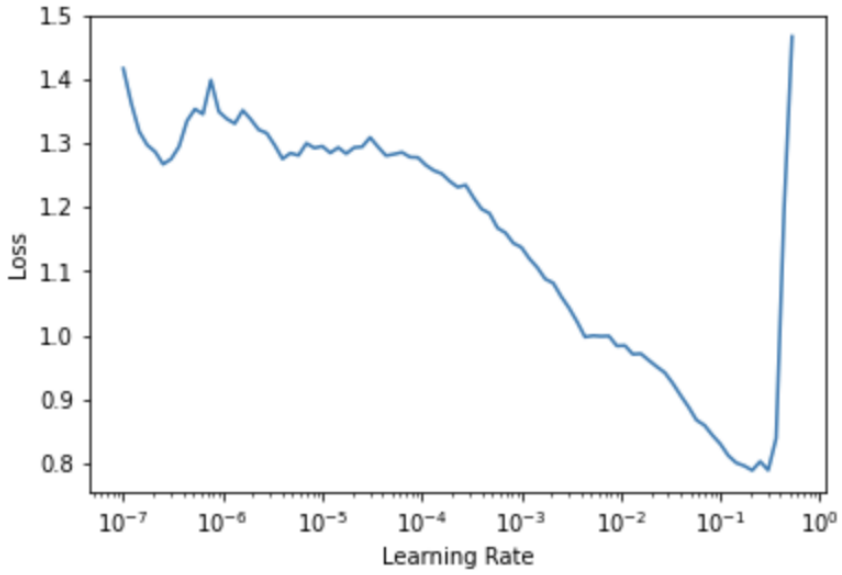


Figure 5: Loss vs learning rate of CNN learner

Discussion

The goal of the project was to create a model capable of accurately acknowledging the presence of pneumonia from only the x-ray images of the chest. While three different classifiers were applied and adapted, CNN was found to provide the best results. This, however, does not conclude that the other algorithms could not be modified to produce better results.

While the kNN classifier was able to produce a relatively high accuracy, the data plot in *Figure 6***,** for the predictions demonstrates that many false positives and negatives were established when predicting the pneumonic cases, where the normal cases are all quite closely concentrated.

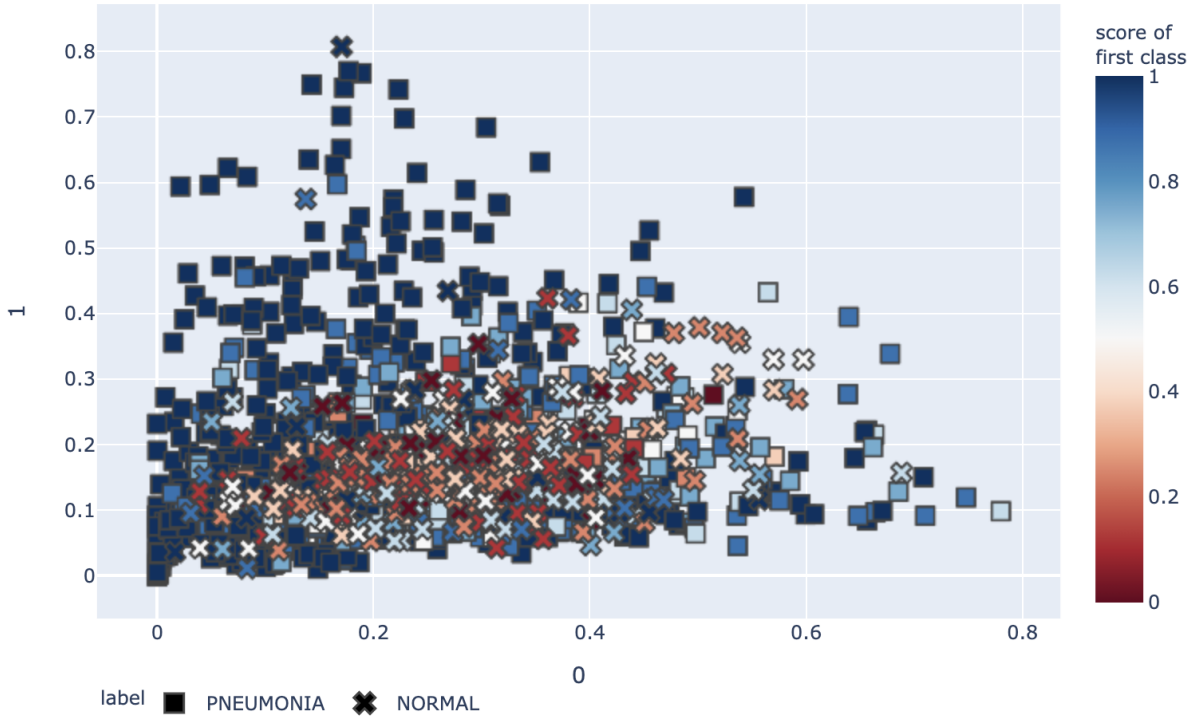


Figure 6: Scatter plot for kNN prediction probabilities on histogram images

Since SVM is a more mathematically complex analysis, the time allocated for the project was not large enough to properly optimize the parameters of function, such as the learning rate, regularization and weight calculation. If properly managed, this classifier has the potential to drastically increase its resulting accuracy.

Conclusions

After running kNN, SVM and CNN classifiers on the dataset of 5865 chest x-ray images, it was concluded that the CNN model provided the best results with an accuracy of 96%. While kNN and SVM were deemed insufficient, it cannot be concluded that these classifiers cannot be

modified to produce better results. The goal of this paper was to create an accurate model to determine the presence of pneumonia in the lungs, and therefore all work concluded with CNN.

Methods

kNN

The first image classifier used was k-Nearest Neighbour (kNN). While this method is the simplest of all machine learning models, since in fact a model is not created, it has been proven to be effective in computer vision. The methodology of this algorithm can be summarized with the following: Tell me who your neighbours are and I will tell you who you are.

To apply our data, all images were extracted from their relative folders and classified by the title of their relative folder name. In this way our data was already separated in the three major sets: train, test and validate, where each contained images labelled as with pneumonia and normal. Each image was then loaded into two different arrays to test the results of two styles of data: raw images and histograms. Once loaded, the test images were first normalized and then run through a cross validation score function to determine the optimal value for k, which represents the number of neighbours taken into account when predicting the class of any given image. *Figure 7* shows the results of this test.

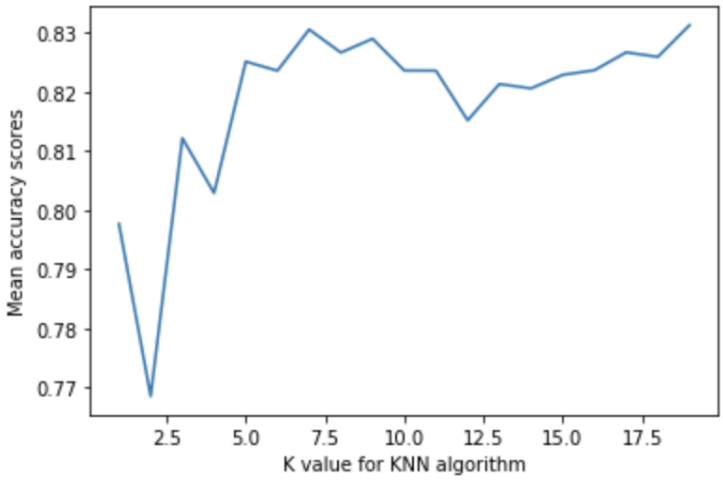


Figure 7: Cross validation score of k values from 1 to 20

As a result, a k value of 7 was chosen and applied to two classifiers, one for the raw images and one for the histograms. The accuracies of each were 91.18% and 84.28%, respectively. These results were confirmed by running several tests with different k values, and we can see in *Figure 8* that our calculations were confirmed.

Run #	k	Raw Pixel Acc	Histogram Acc
1	3	91.03%	83.28%
2	1	90.95%	81.67%
3	7	91.18%	84.28%
4	9	90.72%	83.66%

Figure 8: Several kNN runs validating selected k value

While accuracy, being the total correct predictions divided by the total predictions, is a very useful metric, several others were taken into account to validate the effectiveness of the

function. The confusion matrix, precision, recall, f1-score and support for both classifiers were calculated. In every case, the raw images ranked higher than the histograms.

SVM

The following algorithm used and tested was the Support Vector Machine (SVM). For this algorithm, the images were loaded with the colour channels taken into account. Later, the classifier is trained through stochastic gradient descent, which calculates the loss of the prediction and uses the gradient to determine which direction to move in the following iteration to find the local minimum in the loss function.

The training and validation accuracy of SVM were calculated to be 70.1% and 68.3%, respectively, with an AUC of 0.53, which can be seen in *Figure 3*. These results are quite low and not sufficient to provide an effective prediction.

CNN

A Convolutional Neural Network (CNN) is a deep learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While filters are hand-engineered in primitive methods, with enough training, ConvNets have the ability to automatically learn these filters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of neurons in the human brain and was inspired by the organization of the Visual Cortex.

The CNN created for this project is inspired by the fast.ai library, which comes with a wide range of in-built functions. After organizing the data in a Datablock, the model was trained for 3 epochs and produced accuracies of 96%. In *Figure 9*, we can see the confusion matrix, which is a great improvement from the previous classifiers.

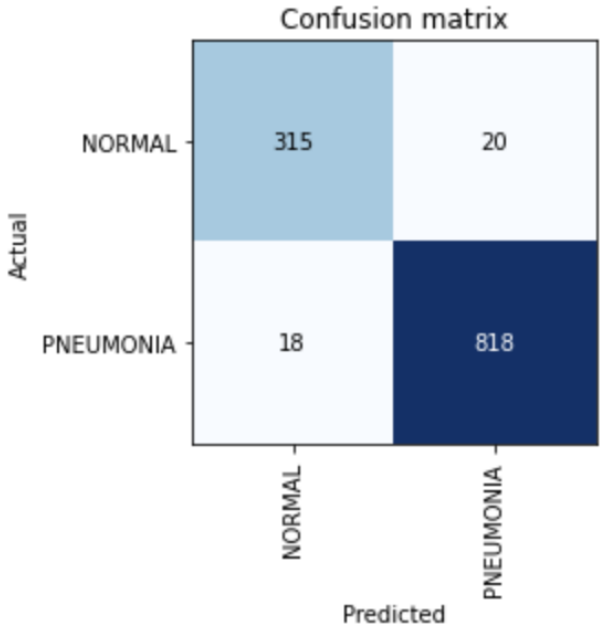


Figure 9: Confusion matrix of CNN model

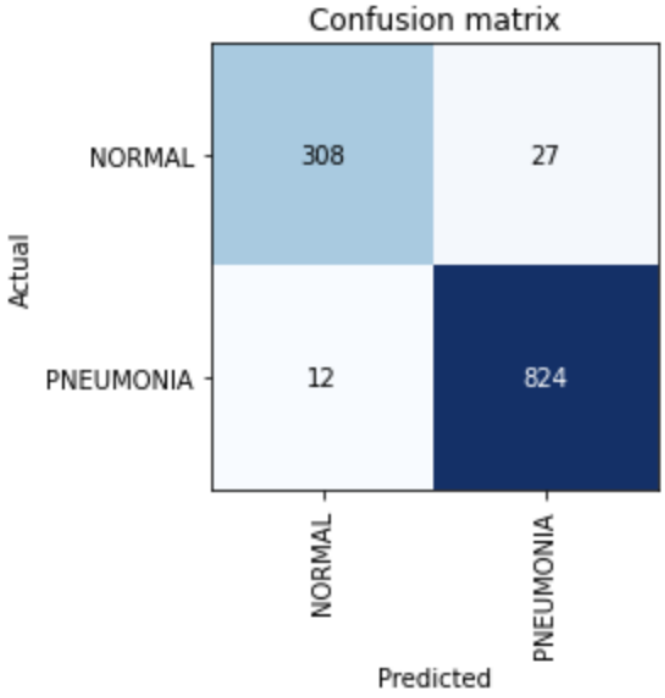
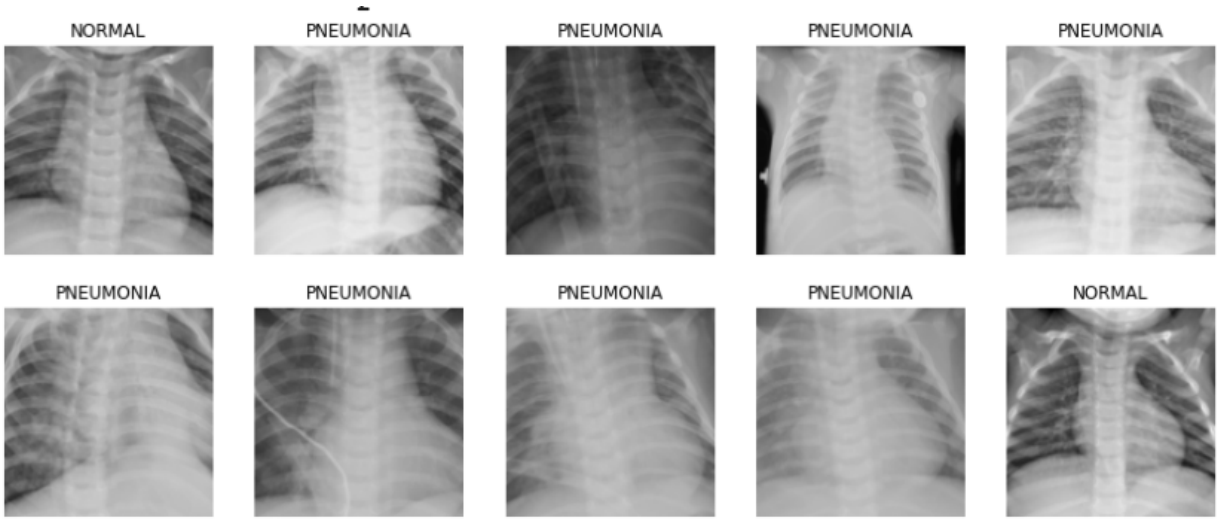


Figure 10: Confusion matrix of CNN model with discriminative learning rates and unfreezing model

Later, the model was modified by adding discriminative learning rates and applying the unfreezing model to test whether any noticeable differences were achieved. As seen in *Figure 10*, the first model was very accurate and no other modifications were applied.

Appendix



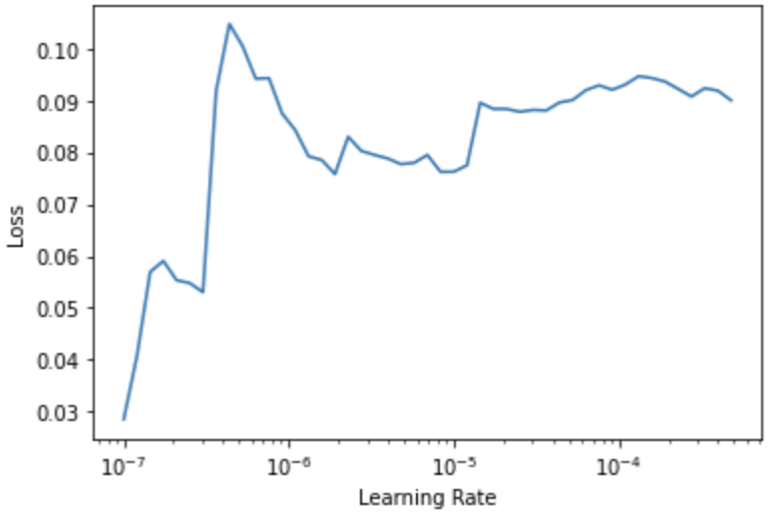
Sample from dataset with classes

	Normal	Pneumonia
Normal	271	86
Pneumonia	16	931

Confusion matrix kNN for raw images

	Normal	Pneumonia
Normal	262	95
Pneumonia	112	835

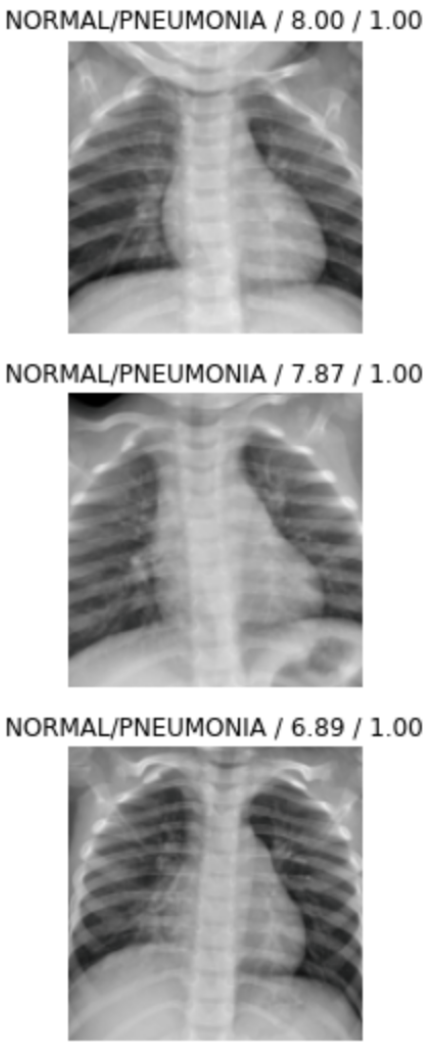
Confusion matrix kNN for histograms



Learning rate vs loss for CNN with discriminative learning rates and unfreezing model

epoch	train_loss	valid_loss	error_rate	accuracy	time
0	0.193305	0.183481	0.054654	0.945346	07:47
1	0.145744	0.150788	0.056362	0.943638	07:49
2	0.105605	0.087236	0.033305	0.966695	07:44

Metrics for CNN model with discriminative learning rates and unfreezing model



Visual of CNN losses