

Managing ML data and Deep Learning models for CT-scans Classification

Ran Tao

Master of Science in Applied Data Science

(Dated: December 4, 2021)

In my final project, I first used CNN (Convolutional Neural Network) to build an image classifier, that can be used to sort chest CT-scans into three categories, namely, the healthy patients, patients with some pre-existing conditions, and patients with various, serious lung conditions that require immediate attention. Then I built an Intuitive UI for non-technical users, which in this case, is administrators of an hospital, so that these users can log in the system, upload an image, get the classification result of this image, and then show all the image information in the database. In this project, I stored all the image inside Firebase Storage, and all the information of an image, including url, some of its metadata and the classification result inside Firebase Firestore Database.

I. DEEP LEARNING PART

A. Data exploration

The dataset used for this project are chest CT scans, as shown in Figure 1. And this data set is composed of two parts, images and labels. There are 13260 images in this data set and the shape of each image is $64 \times 64 \times 1$. The labels data has three kinds of labels, 0, 1, and 2. The numbers of each label are: 0: 10506, 1: 2372, 2: 382.

B. Data Preprocessing

First, I used `train_test_split` methods from `sklearn.model_selection` to split the data set into three parts, training data, validation data, and test data. The ratio of these three parts are 0.6, 0.2, 0.2.

Secondly, because the labels are super imbalanced, I decided to use data augmentation to balance the training data set. I imported `ImageDataGenerator` from `keras.preprocessing.image`, and used this function to create a image generator, which create new images from a given image by rotating, zooming, shearing....Then I used images whose labels are 1 to create many images so that the number of images with label 1 is equal to the number of images with label 0. Then I repeat the same process to create many images with label 2 so that the number of images with label 2 is equal to the number of images with label 0. By doing so, I created a data set with balanced labels.

C. Model Selection

Firstly, I used a CNN with only one convolutional layer, the kernel size is 3×3 , the strides are 1, the padding are valid, and activation function is `relu`. Then I used max pooling and add a Dropout to prevent overfitting. Then I used a fully connected neural networks with a hidden layer and a output layer to output the result of classification. The activation function of these two layers are `relu` and `softmax`.

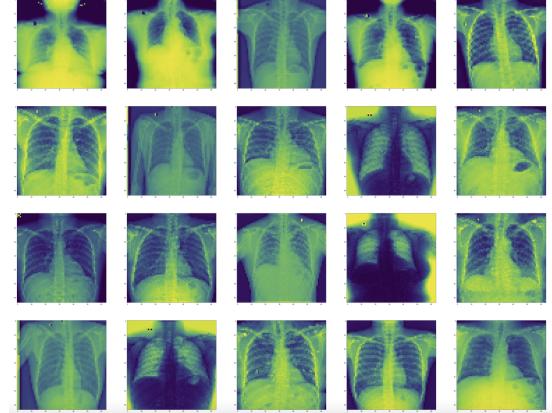


FIG. 1. Example of dataset used for the project

Then I used another CNN with two convolutional layers. The layers are the same as before. After each convolutional layers, I used a max pooling and a dropout layer. Then I used a fully connected neural networks with a hidden layer and a output layer to output the result of classification. The activation function of these two layers are `relu` and `softmax`.

I used training data to train these two models, and I used validation data to measure the performance. Then I used early-stopping to decide when to stop. I computed confusion matrix as well as other evaluation score of these two models. The first model is Fig.2, the second model is Fig.3. As we can see in the pictures, the performance of second model is way better than first model, so I decided to use the second model to classify the images.

II. UI PART

When we enter our system, we can see an UI shown in Figure 4. After the administrator enter the user name and password, he/she can upload an image into our system, as shown in Figure 5. Then when he/she click the upload button, first he/she will see the preview of this image, as shown in Figure 6. Then this image is stored in Firebase Storage, as shown in Figure 7. Then the in-

	precision	recall	f1-score	support
0	0.900	0.721	0.801	2111
1	0.364	0.668	0.471	455
2	0.032	0.047	0.038	86
accuracy			0.690	2652
macro avg	0.432	0.479	0.437	2652
weighted avg	0.780	0.690	0.719	2652

FIG. 2. Confusion Matrix as well as other evaluation score of first model

	precision	recall	f1-score	support
0	0.887	0.923	0.905	2111
1	0.688	0.479	0.565	455
2	0.051	0.081	0.063	86
accuracy			0.820	2652
macro avg	0.542	0.495	0.511	2652
weighted avg	0.825	0.820	0.819	2652

FIG. 3. Confusion Matrix as well as other evaluation score of second model

formation of this image, including its url, metadata and the classification result of this image in stored at Firebase Cloud Firestore, as shown in Figure 8.

Then when the user click the button named Classify the severity of the patient's condition, he/she can get the classification result of this image, as shown in Figure 9. Then when he/she click the button 'See all the data in the database', he/she will get all of our data in the database, as shown in Figure 10.



FIG. 4. Login in page of our system

III. CONCLUSIONS

In my final project, I first used CNN (Convolutional Neural Network) to build an image classifier, that can be used to sort chest CT-scans into three categories, namely, the healthy patients, patients with some pre-existing conditions, and patients with various, serious lung conditions that require immediate attention. Then I built an Intuitive UI for non-technical users, which in this case, is administrators of an hospital, so that these users can log in the system, upload an image, get the classification result

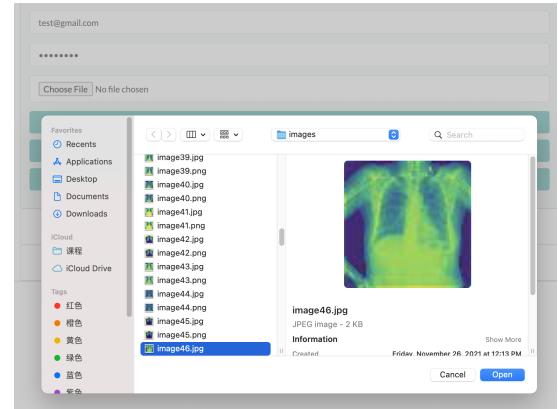


FIG. 5. Upload an image

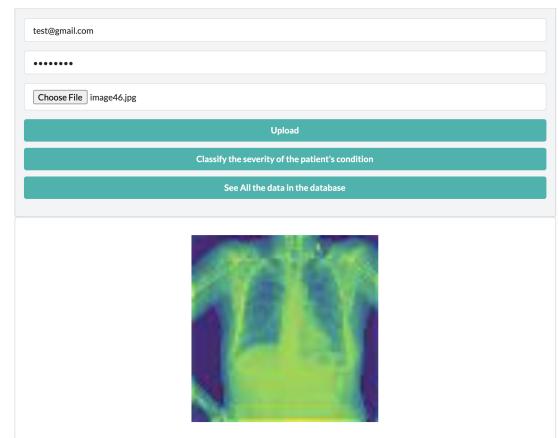


FIG. 6. Preview this image

Storage			
docs51-demo-2	image41.jpg	1.47 KB	image/jpeg Nov 30, 2021
	image42.jpg	1.74 KB	image/jpeg Oct 2, 2021
	image43.jpg	1.59 KB	image/jpeg Nov 30, 2021
	image44.jpg	1.59 KB	image/jpeg Nov 30, 2021
	image45.jpg	1.52 KB	image/jpeg Nov 30, 2021
	image46.jpg	1.79 KB	image/jpeg Nov 30, 2021
	image47.jpg	1.59 KB	image/jpeg Dec 2, 2021
	image48.jpg	1.51 KB	image/jpeg Oct 2, 2021
	image49.jpg	1.7 KB	image/jpeg Nov 29, 2021
	image4a.jpg	1.29 KB	image/jpeg Oct 2, 2021
	image4b.jpg	1.69 KB	image/jpeg Nov 30, 2021
	image4c.jpg	1.61 KB	image/jpeg Oct 2, 2021
	image4d.jpg	1.05 KB	image/jpeg Nov 30, 2021
	image4e.jpg	1.27 KB	image/jpeg Nov 30, 2021
	image4f.jpg	1.39 KB	image/jpeg Nov 30, 2021

FIG. 7. Store this image in Firebase Storage

of this image, and then show all the image information in the database. In this project, I stored all the image inside Firebase Storage, and all the information of an image, including url, some of its metadata and the classification result inside Firebase Firestore Database.

The screenshot shows the Firebase Cloud Firestore interface. A collection named 'images' is selected. Inside, there is a single document named 'image46.jpg'. The document's data is displayed in JSON format:

```

{
  "cla.result": 1644,
  "content-type": "image/jpeg",
  "size": "2021-12-07T01:50:58.850Z",
  "timeCreated": "2021-12-07T01:50:58.831Z",
  "updated": "Healthy",
  "url": "https://firebasestorage.googleapis.com/v0/b/ds051-dmoe-2.appspot.com/o/images%2Fimage46.jpg?alt=media&token=b1a2-e811-ef7f019"
}
  
```

FIG. 8. Store information of an image in Firebase Cloud Firestore

The screenshot shows a web-based application for image classification. It has a file input field labeled 'Choose File' with the file name 'image46.jpg'. Below it is a large green and blue heatmap-style image of a chest X-ray. At the bottom, the text 'Classification result: Healthy' is displayed.

FIG. 9. Get the classification result of the uploaded image

ImageURL	Classification Result	Content Type	Size	Created Time	Updated Time
https://firebasestorage.googleapis.com/v0/b/ds051-dmoe-2.appspot.com/o/images%2Fimage46.jpg?alt=media&token=b1a2-e811-ef7f019	Healthy	image/jpeg	1764	2021-12-07T01:50:58.850Z	2021-12-07T01:50:58.850Z
https://firebasestorage.googleapis.com/v0/b/ds051-dmoe-2.appspot.com/o/images%2Fimage46.jpg?alt=media&token=b1a2-e811-ef7f019	Healthy	image/jpeg	1711	2021-11-20T15:30:00.276Z	2021-11-20T15:30:00.276Z
https://firebasestorage.googleapis.com/v0/b/ds051-dmoe-2.appspot.com/o/images%2Fimage46.jpg?alt=media&token=b1a2-e811-ef7f019	Healthy	image/jpeg	1644	2021-12-07T01:50:44.546Z	2021-12-07T01:50:44.546Z
https://firebasestorage.googleapis.com/v0/b/ds051-dmoe-2.appspot.com/o/images%2Fimage46.jpg?alt=media&token=b1a2-e811-ef7f019	Has pre-existing conditions	image/jpeg	1518	2021-12-07T01:51:44.174Z	2021-12-07T01:51:44.174Z
https://firebasestorage.googleapis.com/v0/b/ds051-dmoe-2.appspot.com/o/images%2Fimage46.jpg?alt=media&token=b1a2-e811-ef7f019	Healthy	image/jpeg	1454	2021-11-20T15:31:53.708Z	2021-11-20T15:31:53.708Z
https://firebasestorage.googleapis.com/v0/b/ds051-dmoe-2.appspot.com/o/images%2Fimage46.jpg?alt=media&token=b1a2-e811-ef7f019	Has pre-existing conditions	image/jpeg	1528	2021-11-20T22:53:44.912Z	2021-11-20T22:53:44.912Z
https://firebasestorage.googleapis.com/v0/b/ds051-dmoe-2.appspot.com/o/images%2Fimage46.jpg?alt=media&token=b1a2-e811-ef7f019	Healthy	image/jpeg	1717	2021-12-07T14:44.55.094Z	2021-12-07T14:44.55.094Z
https://firebasestorage.googleapis.com/v0/b/ds051-dmoe-2.appspot.com/o/images%2Fimage46.jpg?alt=media&token=b1a2-e811-ef7f019	Healthy	image/jpeg	1803	2021-11-20T23:28:02.098Z	2021-11-20T23:28:02.098Z

FIG. 10. Get all the images in the database

CODE AVAILABILITY

Code is available at <https://drive.google.com/drive/folders/1ay6FnnIP01N70VjSvvK2ew6usf9v36Hi?usp=sharing>.