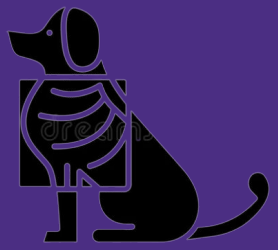
XRayDawgz



XRD Prediction of Crystal Structure by CNN

By: Zhi-Hong Kao, Robert Biegaj, Cheng-yuan Wu, Yu-Hsuan Hsiao

UNIVERSITY of WASHINGTON

Background

All crystalline materials in nature have chemical and physical properties strongly dependent on atomic structures

—>SOLUTION:

to use CNN to predict crystal structure from any XRD pattern

User Stories / Cases

Users:

1. A chemist conducted XRD on an unknown material
2. A material scientist is using powder diffraction to rapidly identify a heterogenous, unknown material, which caused peak overlay.
3. A mechanical engineer who conducted XRD on a sample that returned a pattern containing a substantial amount of impurity peaks.

Case:

- In all cases, the users will be inputting the XRD pattern that they obtained into our CNN model, which will return a predicted crystal structure.

Generating XRD Images

Steps:

1. Gather the cif data from ICSD for each classification.
2. With help of package “xrayutilities” - take cif_data and theoretically predict the powder diffraction angles and intensities
3. Plot these values with matplotlib to generate our dataset of theoretical XRD patterns:
    a. 1330 training (665 for each BCC & FCC)
    b. 196 test/validation (98 for each)

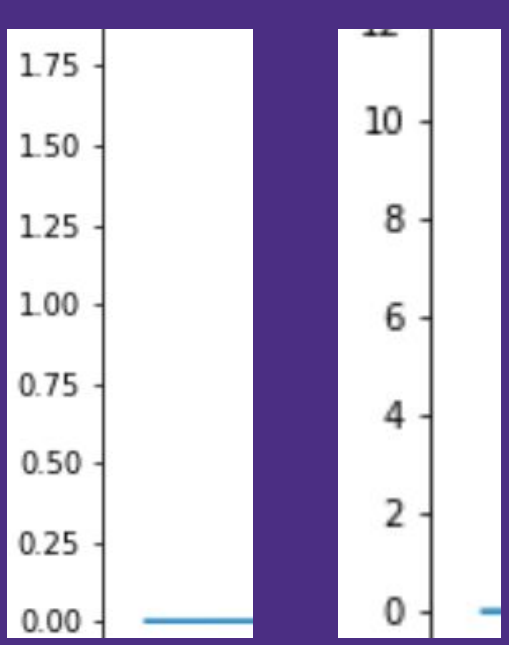
Usage of Software Engineering Tools

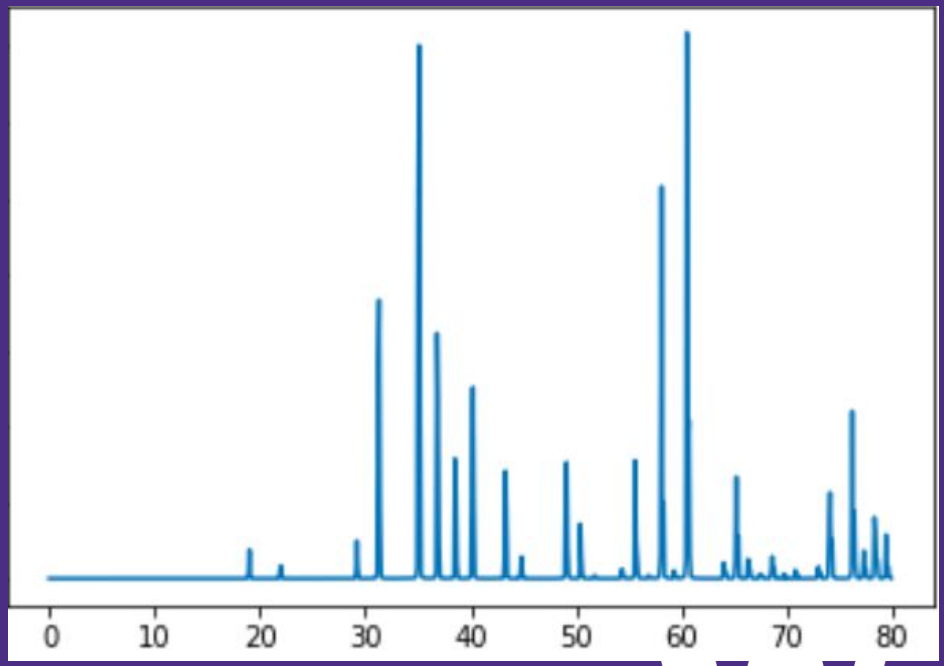
- Using OpenCV to do image preprocessing(cutting)
- CNN model was built by Keras, and using Tensorflow as backend
- Model was training and fitting on Google Colab (which provide free GPU)

Data Preprocessing

The y-axis scale is different for each data

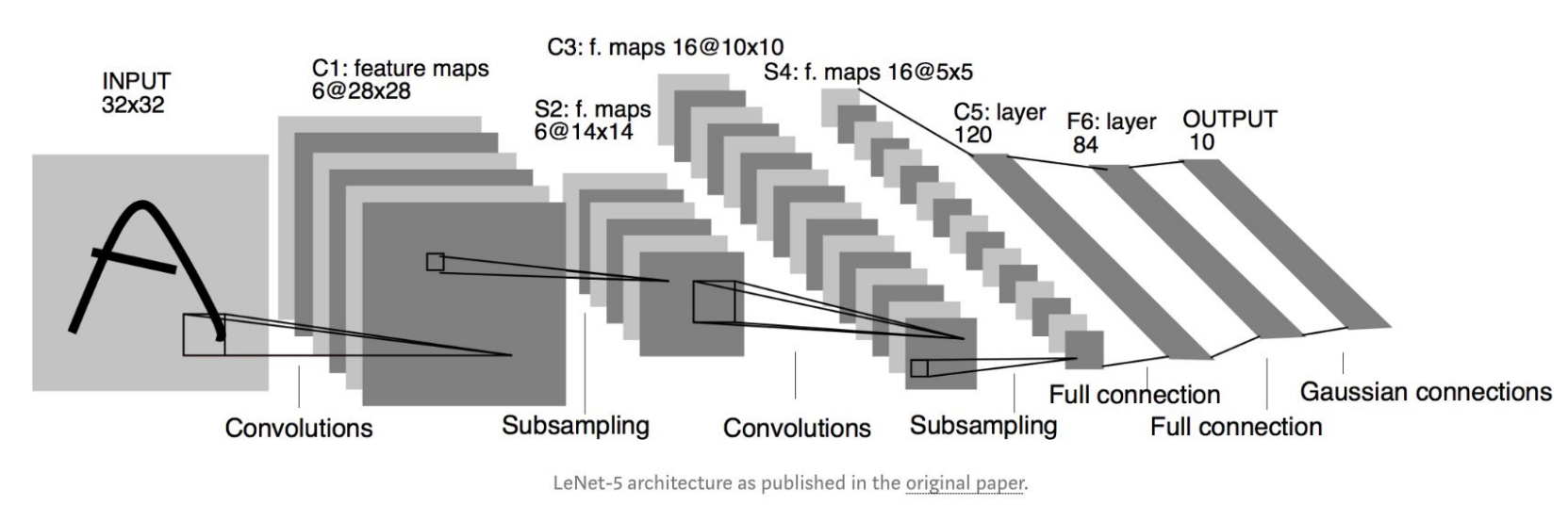
→ Remove the y-axis scale





CNN Model structure

Based on LeNet, and add a extra layer



LeNet-5 architecture as published in the original paper.

3 convolution layers and 3 pooling.

CNN Model structure

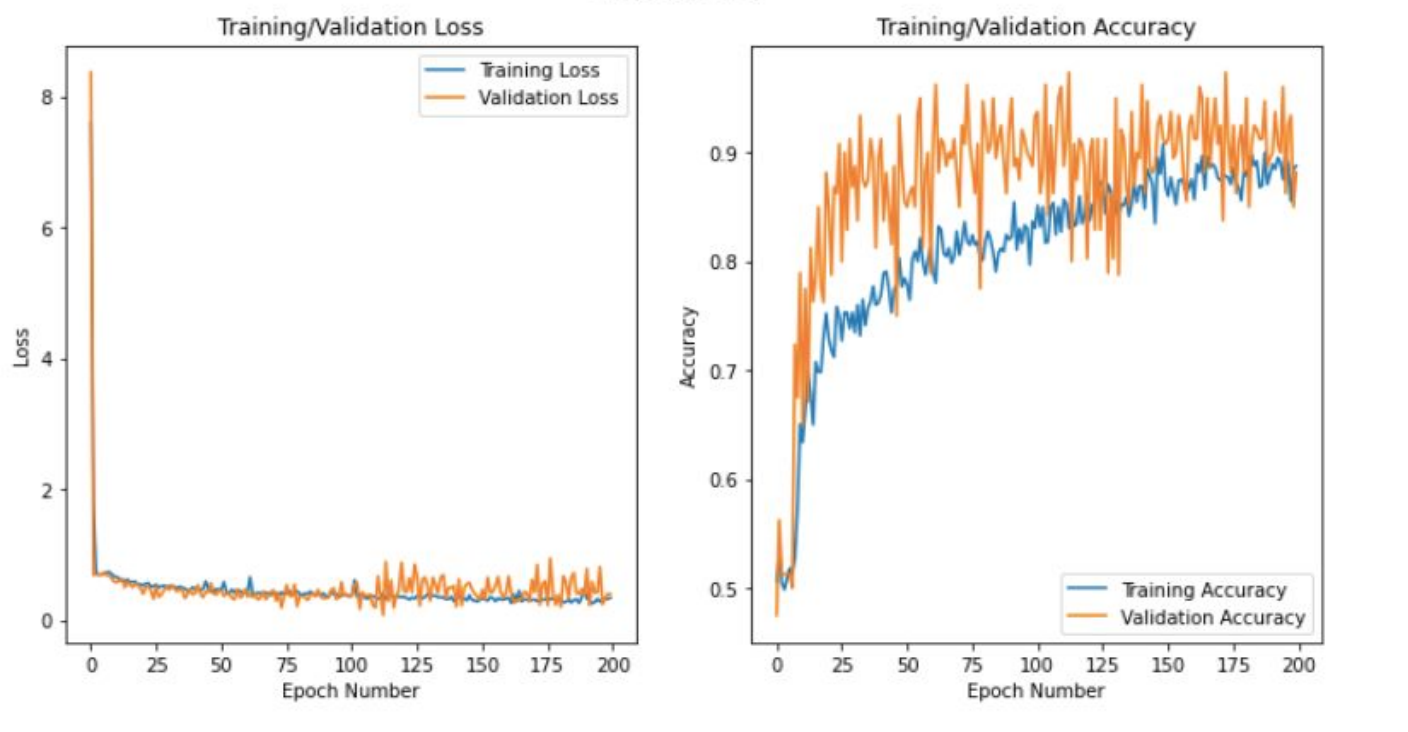
```
# build CNN model
model = Sequential()
# extract features by iterating across image
model.add(Conv2D(32, (3, 3), input_shape=(img_width, img_height, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(32, (3, 3), input_shape=(img_width, img_height, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(64, (3, 3), input_shape=(img_width, img_height, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(ZeroPadding2D((1,1)))
#convert 3D features to 1D feature vectors
model.add(Flatten())
model.add(Dense(64))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(1))
model.add(Activation('sigmoid'))
```

Loss function: binary crossentropy

Optimizer: rmsprop

Epochs: 32 →75 →100→ 125 →200

Results:

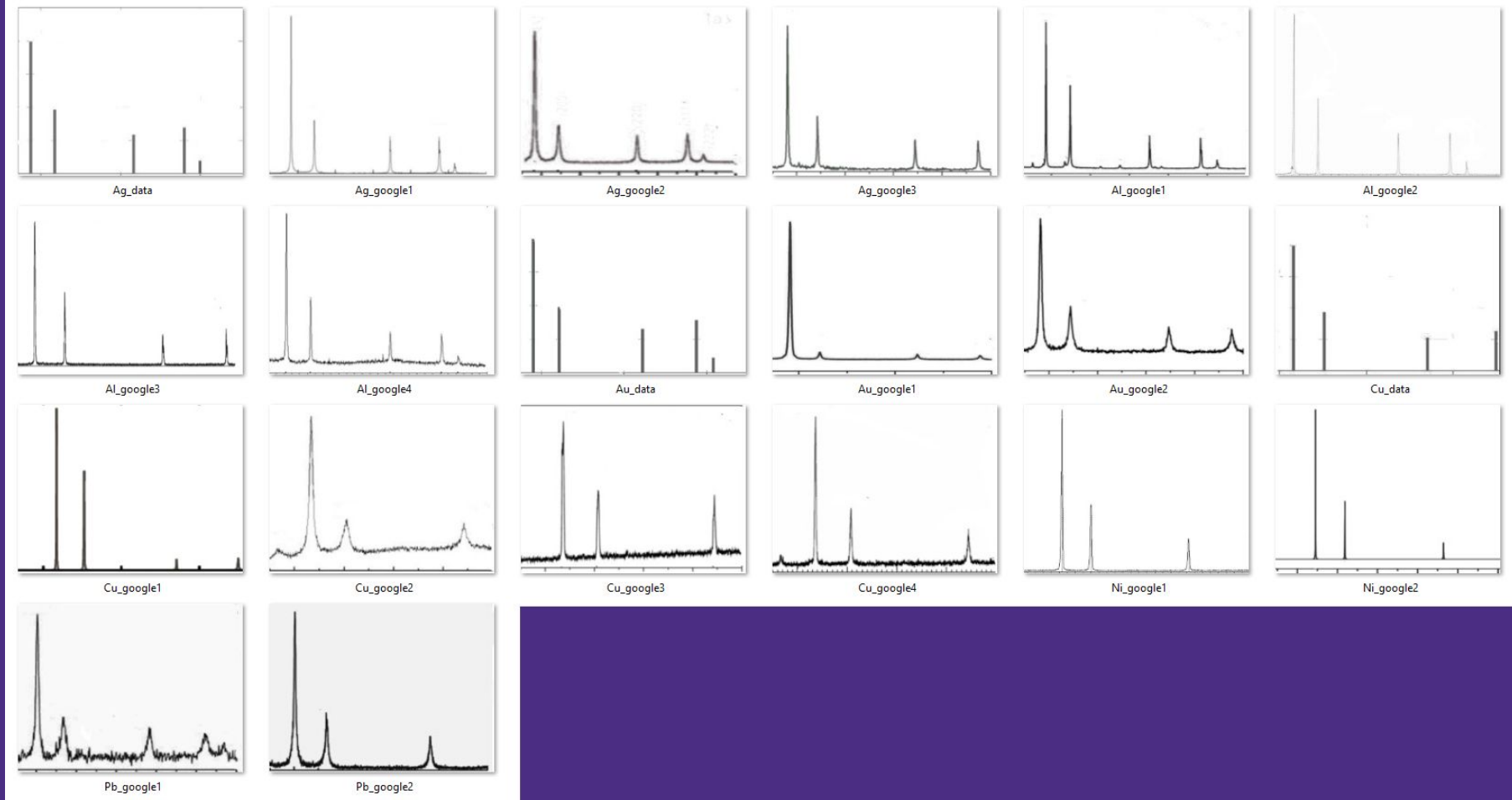


Testing of theoretical validation XRD patterns:

- **BCC prediction accuracy:** 92.86%
- **FCC prediction accuracy:** 86.73%

Demo and Example

Gathered real XRD images - FCC crystal structure



Demo and Example

Run function to predict crystal structure

```
In [1]: import Predict
        Using TensorFlow backend.

In [2]: predict.crystal_structure('path_to_folder_containing_images')
        Ag_data.jpg is FCC!
        Ag_google1.jpg is FCC!
        Ag_google2.jpg is FCC!
        Ag_google3.jpg is FCC!
        Al_google1.jpg is FCC!
        Al_google2.jpg is FCC!
        Al_google3.jpg is FCC!
        Al_google4.jpg is BCC!
        Au_data.jpg is FCC!
        Au_google1.jpg is FCC!
        Au_google2.jpg is BCC!
        Cu_data.jpg is FCC!
        Cu_google1.jpg is FCC!
        Cu_google2.jpg is BCC!
        Cu_google3.jpg is BCC!
        Cu_google4.jpg is FCC!
        Ni_google1.jpg is FCC!
        Ni_google2.jpg is FCC!
        Pb_google1.jpg is BCC!
        Pb_google2.jpg is FCC!
```

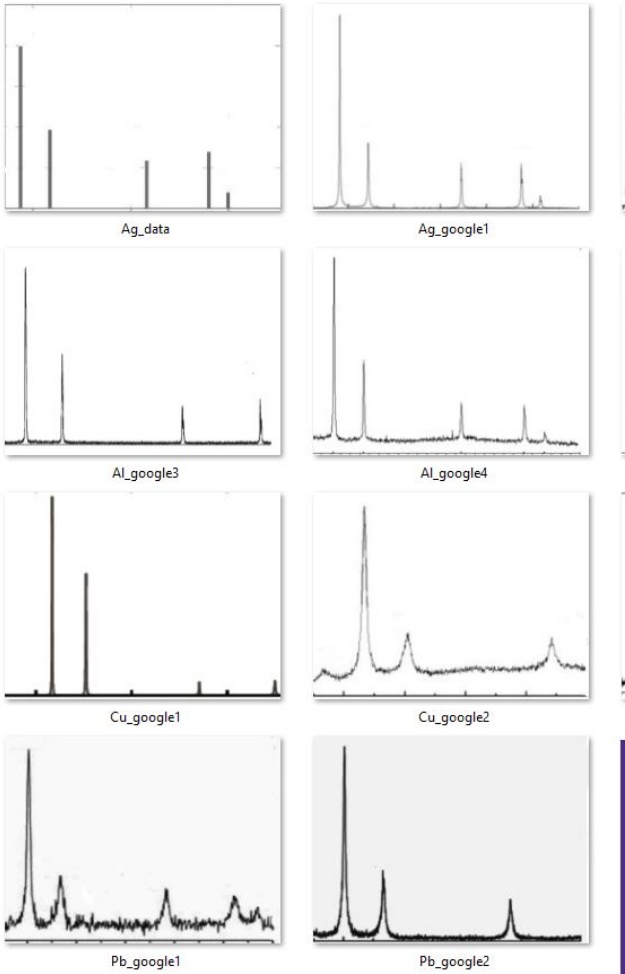
Demo and Example

Run function to predict crystal structure

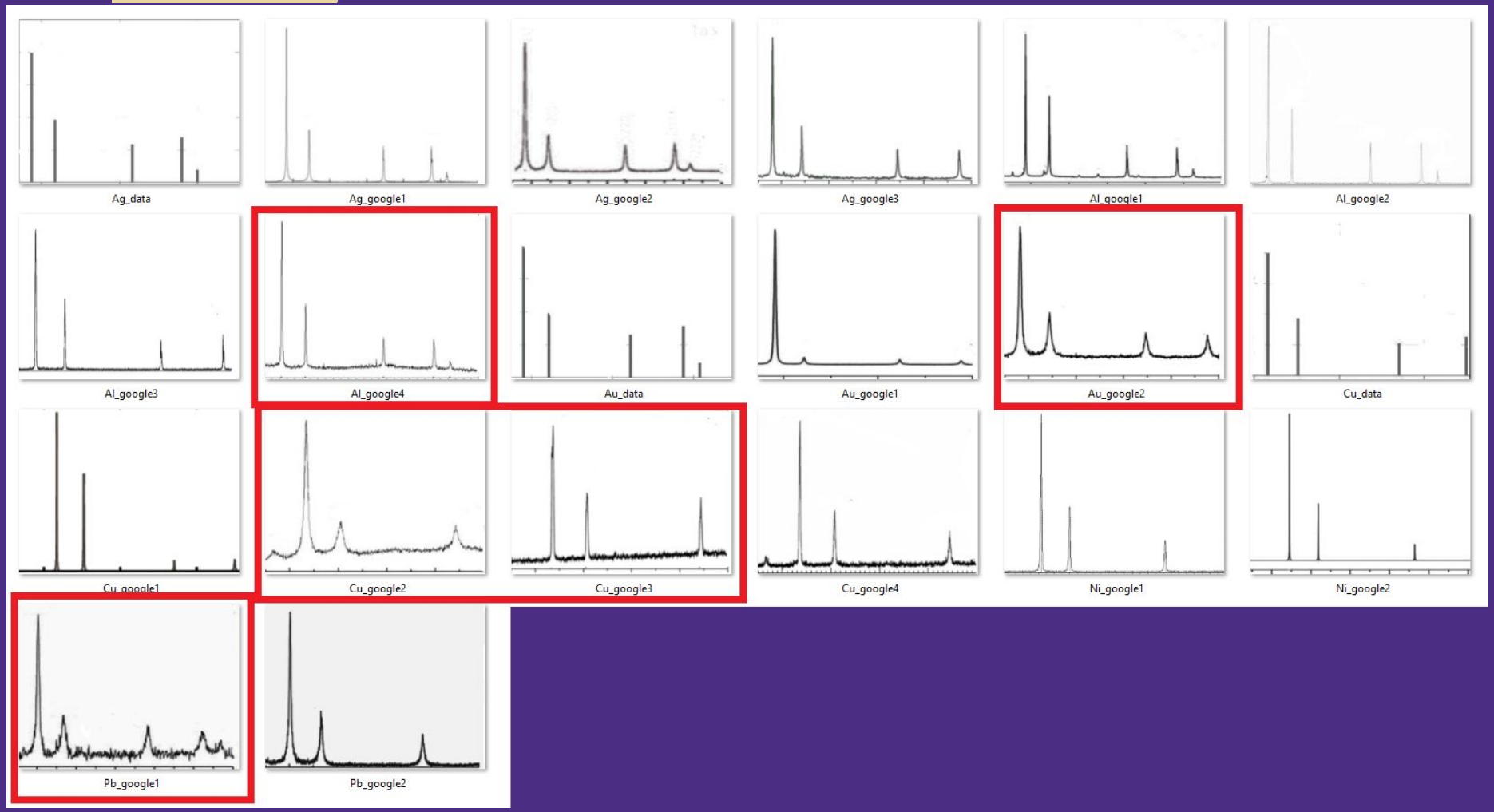
```
In [1]: import Predict
        Using TensorFlow backend.

In [2]: predict.crystal_structure('path_to_folder_containing_images')
        Ag_data.jpg is FCC!
        Ag_google1.jpg is FCC!
        Ag_google2.jpg is FCC!
        Ag_google3.jpg is FCC!
        Al_google1.jpg is FCC!
        Al_google2.jpg is FCC!
        Al_google3.jpg is FCC!
        Al_google4.jpg is BCC!
        Au_data.jpg is FCC!
        Au_google1.jpg is FCC!
        Au_google2.jpg is BCC!
        Cu_data.jpg is FCC!
        Cu_google1.jpg is FCC!
        Cu_google2.jpg is BCC!
        Cu_google3.jpg is BCC!
        Cu_google4.jpg is FCC!
        Ni_google1.jpg is FCC!
        Ni_google2.jpg is FCC!
        Pb_google1.jpg is BCC!
        Pb_google2.jpg is FCC!
```

15/20 = 75%!

Lessons Learned



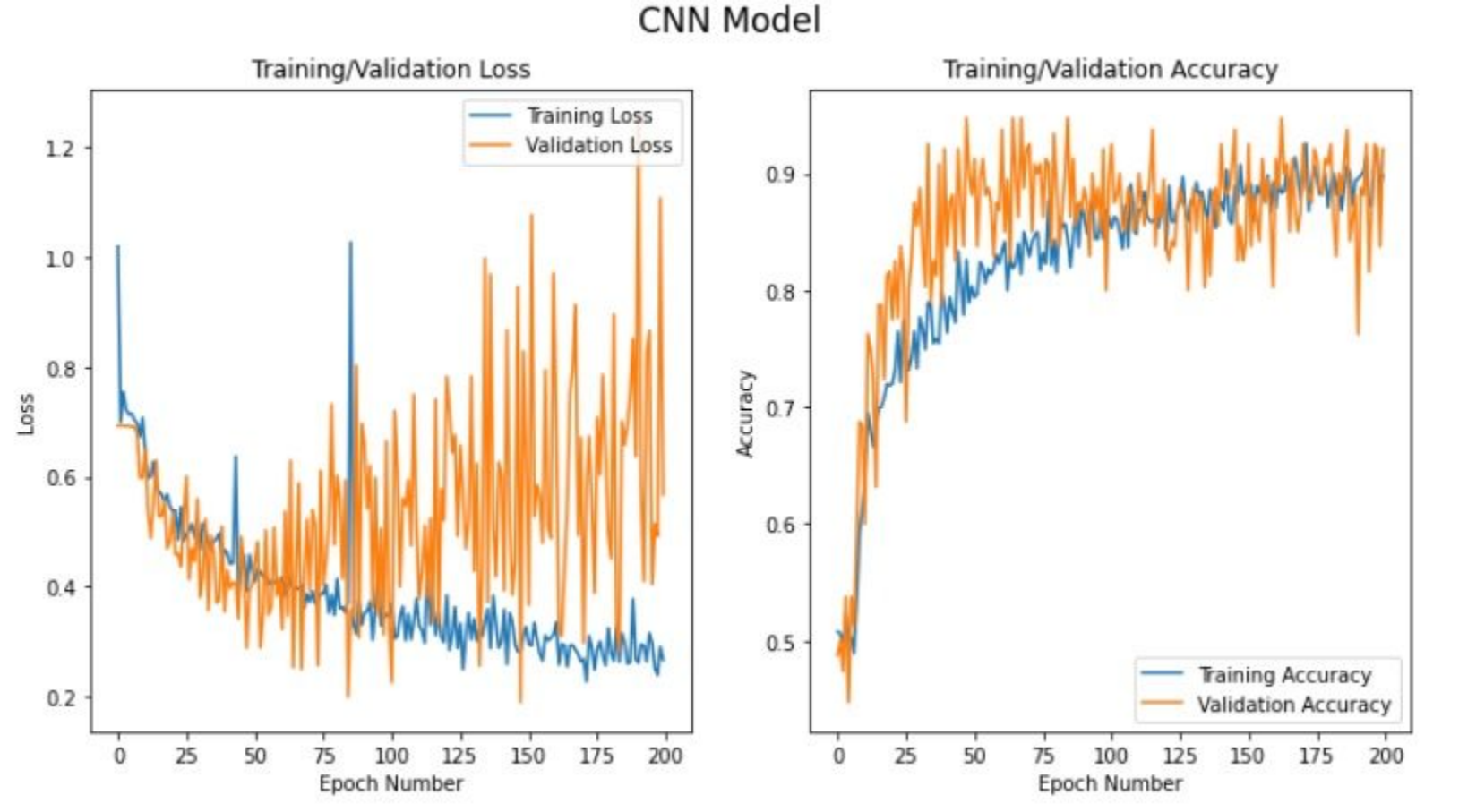
Lessons Learned



Lessons Learned

FURTHER TRAINING AND DEVELOPMENT OF MODEL!

Lessons Learned: Model Training



More Lessons Learned

- Create branches!
- Test driven development and use flake8 more frequently to check for PEP8 compliance

Future Development

- So far, our CNN model is only capable of predicting between Cubic FCC or BCC crystal structures.
    - Expand our model to predict hexagonal, orthorhombic, tetragonal, monoclinic, trigonal, and triclinic crystal structures.

Thank you