Final Project

Lihui Pan(lp2892)

04/28/2021

Preprocessing

Build Models

Train Accuracy

Test Result

- Select slides to use for the project
  - **Train & Validation:** 091、101、084、081、078、075、064
  - **Test:** 110
- Randomly crop patches from slides
  - Size: **150×150**
  - Only keep patches contains less than **50% percent** non-tissue area
- Balance data
  - Keep patches without cancerous **two times** as those with

```
def find_percentage_tissue_pixels(image,width=150,height=150,intensity=0.8):
        im_gray = rgb2gray(image) assert im_gray.shape == (image.shape[0], image.shape[1])
        indices = np.where(im_gray <= intensity) result = list(zip(indices[0], indices[1]))
        percentage = len(result) / float(width * height) * 100
        return percentage
```

- One zoom level
  - Original
  - Data augmentation
- Different zoom levels
  - Directly put different zoom level patches into the model
  - Train different zoom level patches separately

```
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Flatten())
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(1, activation='sigmoid'))
```

Preprocessing

Build Models

Train Accuracy

Test Result

- One zoom level
  - Original
  - Data augmentation
- Different zoom levels
  - Directly put different zoom level patches into the model
  - Train different zoom level patches separately

```
model.compile(optimizer='adam’, loss='binary_crossentropy',metrics=['accuracy'])
```

```
datagen = ImageDataGenerator(rotation_range=40,
                             width_shift_range=0.2,
                             height_shift_range=0.2,
                             shear_range=0.2,
                             zoom_range=0.2,
                             horizontal_flip=True,
                             fill_mode='nearest')
```

Preprocessing

Build Models

Train Accuracy

Test Result

- One zoom level
  - Original
  - Data augmentation
- Different zoom levels
  - Directly put different zoom level patches into the model
  - Train different zoom level patches separately

```
image_input_2 = Input(shape=(150, 150, 3))
vision_model_2 = Sequential()
vision_model_2.add(layers.Conv2D(32, (3, 3), activation='relu'))
vision_model_2.add(layers.MaxPooling2D((2, 2)))
vision_model_2.add(layers.Conv2D(64, (3, 3), activation='relu'))
vision_model_2.add(layers.MaxPooling2D((2, 2)))
vision_model_2.add(layers.Conv2D(128, (3, 3), activation='relu'))
vision_model_2.add(layers.MaxPooling2D((2, 2)))
vision_model_2.add(layers.Conv2D(128, (3, 3), activation='relu'))
vision_model_2.add(layers.MaxPooling2D((2, 2)))
vision_model_2.add(layers.Flatten())
encoded_image_2 = vision_model_2(image_input_2)
```

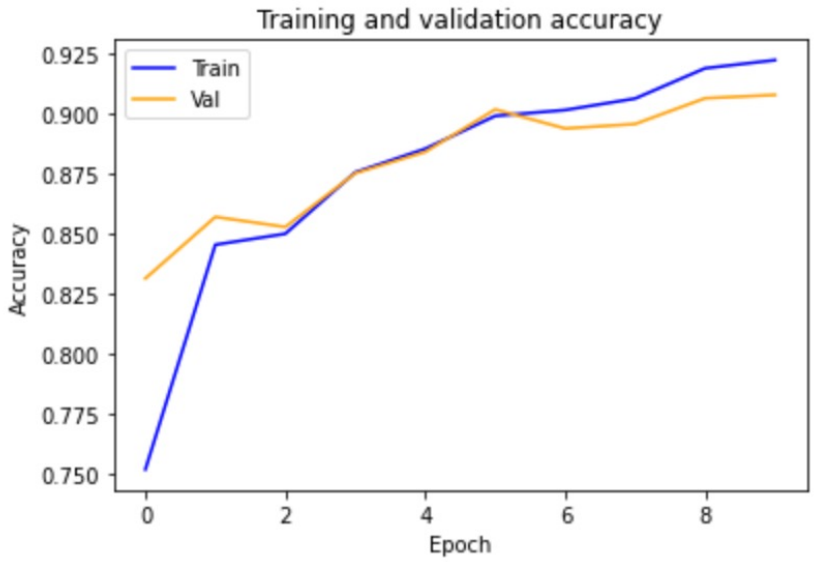
Preprocessing

Build Models

Train Accuracy

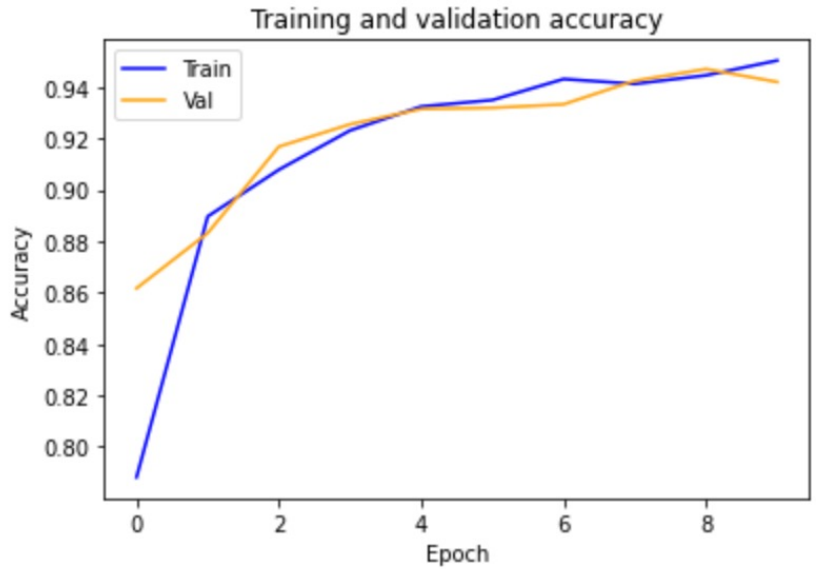
Test Result

Original(90.81%)

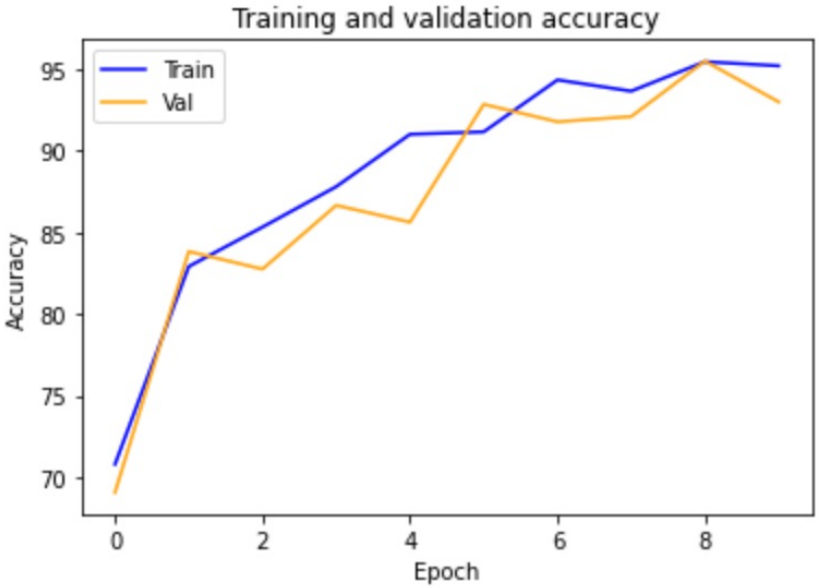


Data augmentation(89.79%)

Directly(94.72%)



Separately(92.48%)



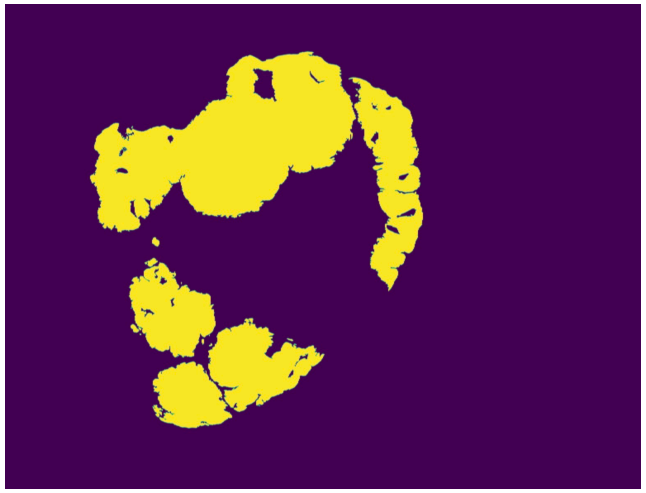
Preprocessing

Build Models

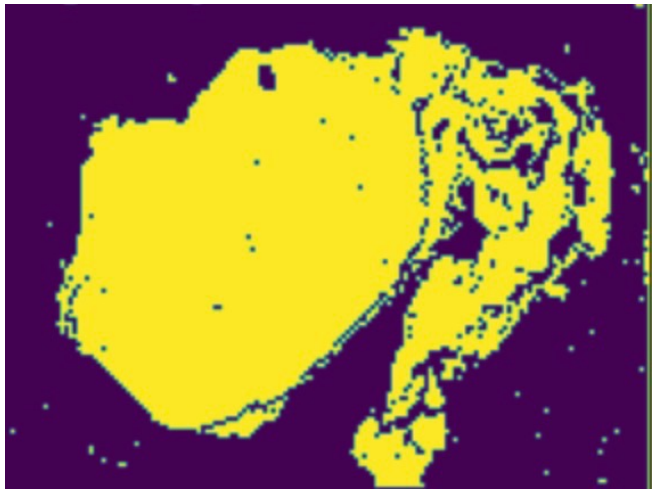
Train Accuracy

Test Result

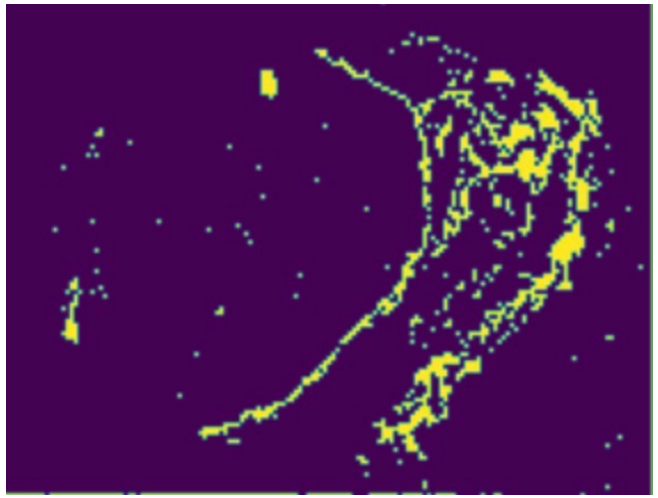
True Map



Original(99%)



Directly(1%)



- The high accuracy of the third model maybe mainly due to the larger dataset.
- The train dataset should be checked again or more carefully, since the final prediction indicates that the train and validation dataset are not representative enough.

Thanks!

Lihui Pan(lp2892)

04/28/2021