



Week 3 Progress Report

Presenter - Matt Behnke

Mentor - Dr. Guo



Goals / Table of Contents

1. Migrating existing code to Google Colab
2. Approach 100% of files being included
3. Visualizing the data matrices
4. Cropping matrices based on melt pool
5. Increasing data instances of bad records
6. Constructing Initial Deep Learning Model



Setting up in Google Colab



Changing to Google Colab

- Very Minimal Differences which was easy!
- Shows access to GPU and its device properties
- Code for reading files in google colab differed from typical operating systems calls in python

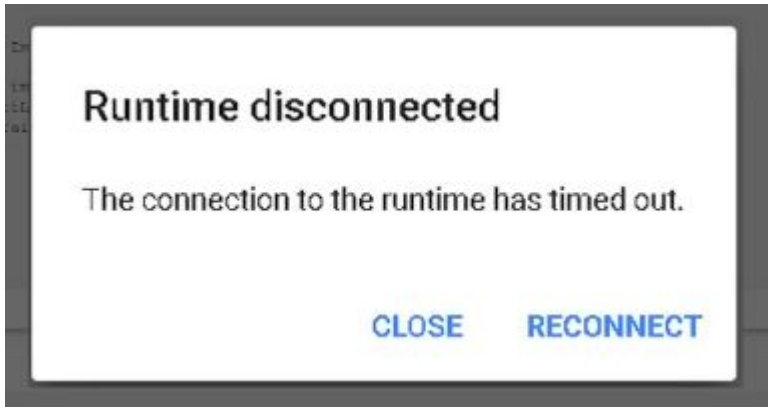
```
incarnation: 17904489025945183401
physical_device_desc: "device: XLA_GPU device", name: "/device:GPU:0"
device_type: "GPU"
memory_limit: 15701463552
locality {
  bus_id: 1
  links {
  }
}
```

```
from google.colab import drive
drive.mount('/content/drive')
path = "/content/drive/My Drive/Colab Notebooks/Porosity_IndexData2.csv"
```

```
for file in glob.glob(path):
    titles = file.split('/content/drive/My Drive/Colab Notebooks/PorosityData/')
```



Google Colab Reconnect



Google Colab would reconnect without regular interaction, which would cause to read files in again (which wasted *a lot* of time)

Found interesting work around code, to 'cheat' its system

```
> function ClickConnect(){  
    document.querySelector("colab-connect-button").click()  
    console.log("Clicked on connect button");  
}  
setInterval(ClickConnect,60000)
```



Including Almost All Data Files



Original Files that were matched

- 1200 files were matched to their respective dataset by using data in Porosity_IndexData2.csv
 - Using `t-time_x-location_y-location_z-location_layer#.csv` formula provided in *Data in Brief*
- 6 Files are will not be included due to an invalid status of -1
 - Unable to tell whether it was supposed to be 0 or 1 so it was invalid
- 358 files not matched
 - Some files are not named accordingly with naming standards
 - A more encompassing formula/algorithm is needed to match them, which will be discussed next

```
Number of files successfully encoded 1200
Number with invalid status (not 0 or 1) 6
Number of files without a corresponding title 358
```

Incorrect Titles Trends - Rounding


Occasionally, the Porosity_IndexData2.csv datafile would round tp, yp, or zp models

- Values > 05 went to 1
- Values =< 04 went to 0

Data Instance from Porosity_IndexData2.csv

	t	tp	x	y	yp	z	zp	layer	status	value		
1117	1116	321	2	0	29	0	20	91	42	good	0	0

Same File in File Directory

 t321p2_x0_y29p04_z20p91_layer42.csv

Incorrect Titles Trends - Truncation


Occasionally, the Porosity_IndexData2.csv datafile would round tp, yp, or zp models

- All Values with a leading 0 and a following number, drops the leading zero
 - Ex: 04 → 4 or 07 → 7

Data Instance from Porosity_IndexData2.csv























	t	tp	x	y	yp	z	zp	layer	status	value			
25	24	10	6	0	30	97	0	51	2	bad	1	0.277	1

Same File in File Directory

 t10p06_x0_y30p97_z0p51_layer2.csv



Incorrect Title Trends - Omitting zp

 t0p1524_x0_y1p936_z0_layer1.csv	12/7/201
 t0p3049_x0_y3p872_z0_layer1.csv	12/7/201
 t0p4573_x0_y5p808_z0_layer1.csv	12/7/201
 t0p6097_x0_y7p744_z0_layer1.csv	12/7/201
 t0p7622_x0_y9p680_z0_layer1.csv	12/7/201
 t0p9146_x0_y11p62_z0_layer1.csv	12/7/201
 t1p067_x0_y13p55_z0_layer1.csv	12/7/201
 t1p219_x0_y15p49_z0_layer1.csv	12/7/201
 t1p372_x0_y17p42_z0_layer1.csv	12/7/201
 t1p524_x0_y19p36_z0_layer1.csv	12/7/201
 t1p677_x0_y21p30_z0_layer1.csv	12/7/201
 t1p829_x0_y23p23_z0_layer1.csv	12/7/201
 t1p982_x0_y25p17_z0_layer1.csv	12/7/201
 t2p134_x0_y27p10_z0_layer1.csv	12/7/201
 t2p287_x0_y29p04_z0_layer1.csv	12/7/201
 t2p439_x0_y30p97_z0_layer1.csv	12/7/201
 t2p591_x0_y32p91_z0_layer1.csv	12/7/201
 t2p744_x0_y34p85_z0_layer1.csv	12/7/201
 t2p896_x0_y36p78_z0_layer1.csv	12/7/201
 t3p049_x0_y38p72_z0_layer1.csv	12/7/201
 t3p201_x0_y40p65_z0_layer1.csv	12/7/201
 t3p354_x0_y42p59_z0_layer1.csv	12/7/201



Algorithm to Include more files + Results

Algorithm

For every title from the original 358 files not having a matching dataset

1. If $0 \leq t \leq 3$, do not include a zp value and add to successful dataset
2. Else, split title by '_' into t#tp# , x# , y#yp# , z#zp# , layer#
 - a. For t#tp# , y#yp# , and z#zp# , round tp# , yp# , and zp#
 - b. Re-join title with a '_'
 - c. Hash Title in TitleStatusMap, if no error add to successful dataset
 - i. If error occurs, truncate tp# , yp# , and zp#
 - ii. Re-join title with a '_'
 - iii. Hash Title in TitleStatusMap, if no error add to successful dataset
 - iv. If error arises, note that file was unsuccessful

Results

```
Total titles fixed = 354
Here are the list of fixed titles
t321p2_x0_y29p0_z20p91_layer42.csv
.....
```

354 of 358 files were successfully included from the new algorithm

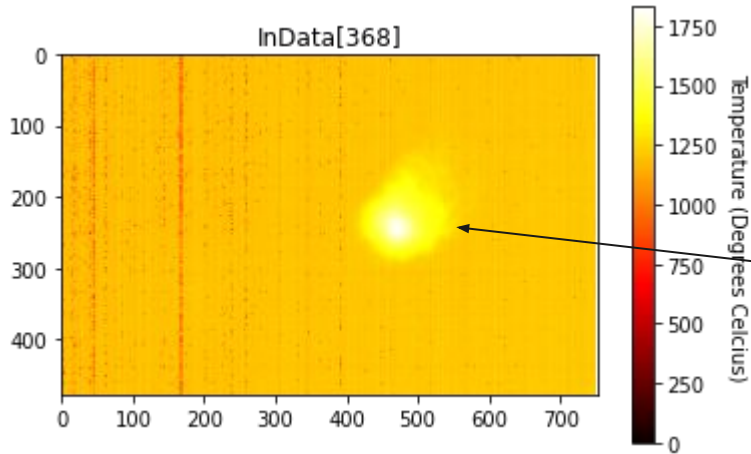
```
Here are the list of titles that were __NOT__ fixed
t0p0000_0_0p0000_0_layer1.csv
t9p908_x0_y29p04_z0p51_layer2.csv
t28p05_x0_y42p59_z1p53_layer4.csv
t32p77_x0_y0p0000_z2p04_layer5.csv
```

The remaining 4 were added manually, by changing values in Porosity_IndexData2.csv to match



Visualizing Data Matrices

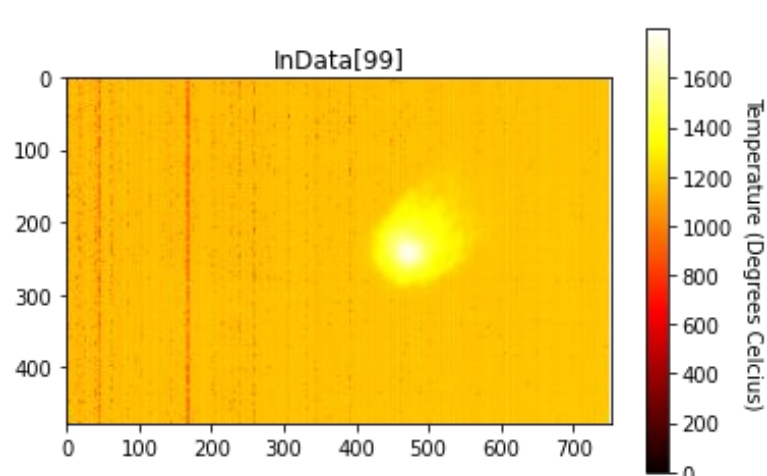
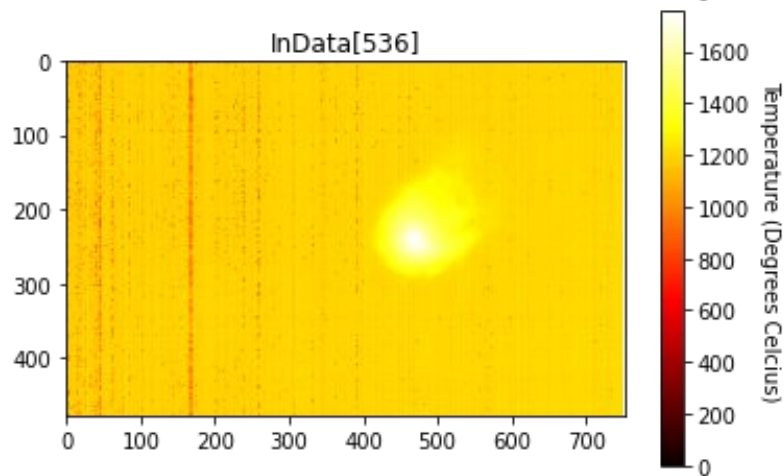
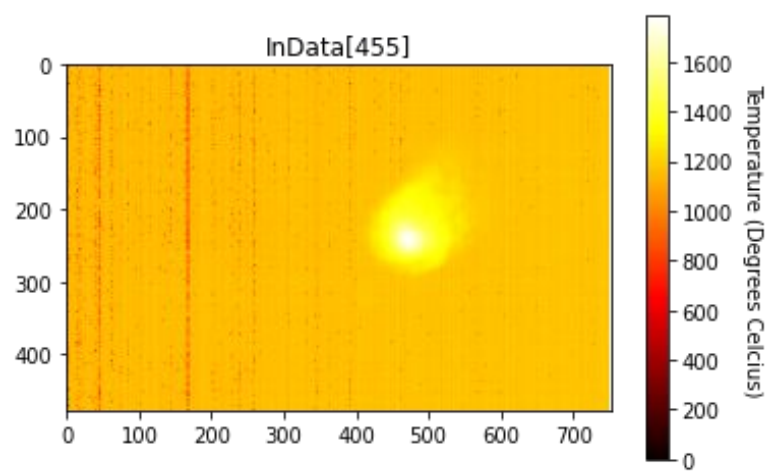
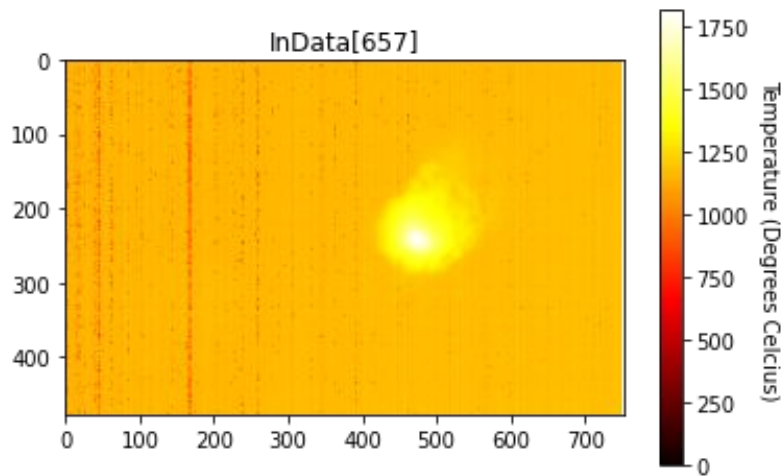
Generating a heatmap of a data matrix



```
import random
x = random.randint(0, len(inData))
fig, ax = plt.subplots()
ax.set_title("InData[{}].format(x)
im, cbar = heatmap(inData[x], None, None, ax=ax, cmap="hot", cbarlabel="Temperature (Degrees Celcius)")
fig.tight_layout()
plt.show()
```

- Example of what the melt pool looks like
- Want to plot multiple random data instances, to find area where to crop

Visualizations of Random Instances of the Data Matrices

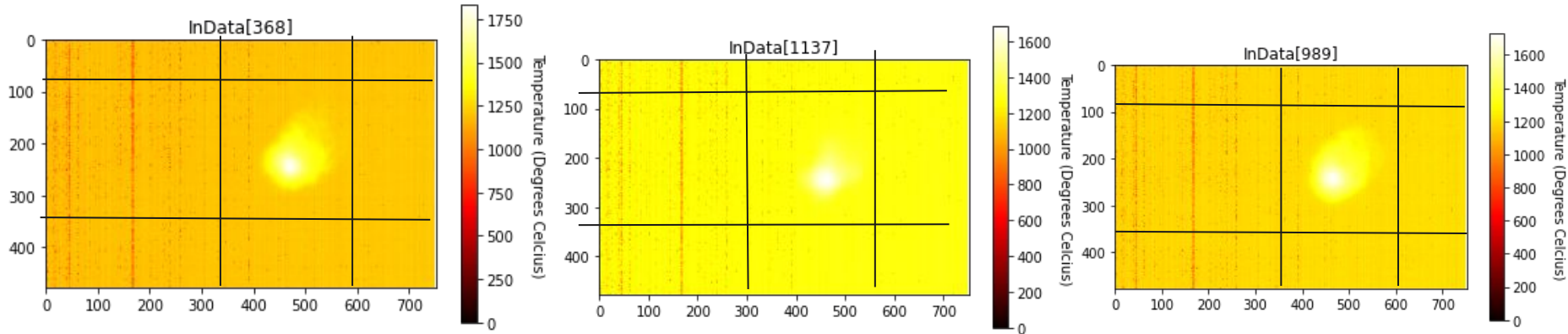




Cropping Matrices based on Melt Pool

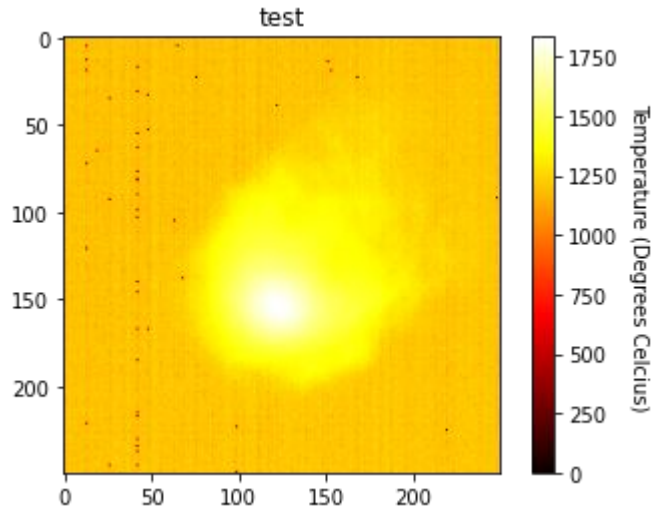


Estimating an encompassing crop



- A region of $90 \leq y \leq 340$ and $350 \leq x \leq 600$ appears to encompass both smaller and larger melt pools
- Lowers total data points from 562,375,680 ($480 \times 752 \times 1558$) to 97,375,000 ($250 \times 250 \times 1558$)

Visualizing the New Crop Boundaries



```
# Test to show visualizing the cropping
test = inData[0][90:340, 350:600]
print(test.shape)
fig, ax = plt.subplots()
ax.set_title("test")
im, cbar = heatmap(test, None, None, ax=ax, cmap="hot", cbarlabel="Temperature (Degrees Celcius)")
fig.tight_layout()
plt.show()
```



Increasing Data Instances of Defects



Increasing Instances of Bad Status

Methodology

While #ofBadInstatnces < #GoodInstances:

Pick Random number (in range of bad instances)

Find corresponding data and result

Add to cropInData and OutData

```
import random
# Create Bad Status List
badStatusList = []
goodStatusListLen = 0
for i in range(len(outData)):
    if(outData[i] == 1):
        badStatusList.append((cropInData[i], 1))
    else:
        goodStatusListLen += 1

tempBadStatusListLen = len(badStatusList)
print("Amount of Bad Instances = {}".format(tempBadStatusListLen))
print("Amount of Good Instances = {}".format(goodStatusListLen))

# Pick a random instance and add it to the crop data list
while tempBadStatusListLen < goodStatusListLen:
    randNum = random.randint(0, len(badStatusList)-1)
    tempInstance = badStatusList[randNum]
    cropInData.append(tempInstance[0])
    outData.append(tempInstance[1])
    tempBadStatusListLen += 1

print("Length of all data now {}".format(len(cropInData)))
```

```
Amount of Bad Instances = 1154
Amount of Good Instances = 1154
Length of all data now 2308
```



Constructing Initial Deep Learning Model



Running out of RAM

Had to cut down on some of the training model input dimensions to not overload the RAM on Google Colab

Interesting work arounds on shape, which I believe cut down on accuracy

```
ResourceExhaustedError                                Traceback (most recent call last)
<ipython-input-49-bfea58214237> in <module>()
     11     return model
     12
--> 13 model = initializeModel()

----- 12 frames -----
/usr/local/lib/python3.6/dist-packages/six.py in raise_from(value, from_value)

ResourceExhaustedError: OOM when allocating tensor with shape[100937500,100] and type float on /job:localhost/replica:0/task:0/device:GPU:0 by allocator GPU_0_bfc [Op:RandomUniform]
```

SEARCH STACK OVERFLOW



Initial Test Model

Show Google Colab Notebook

```
▶ results = model.evaluate( np.expand_dims(xTest, -1), np.expand_dims(yTest, -1), batch_size=128)  
print('test loss, test acc:', results)  
predictions = model.predict(np.expand_dims(xTest, -1))
```

```
↳ 693/693 [=====] - 0s 553us/step  
test loss, test acc: [0.6932741235000919, 0.4935064911842346]
```



Future Steps / Feedback or Reconsiderations

- Appears like (hopefully) most of the pre-processing of data is done, so it's time to get more detailed and intricate with model architecture
- More clever ways to crop than based off visuals
- Office of Advanced Research Computing (ORAC)?
 - I have not heard back =(