

EXP1a: UNET COMPARISONS: 3 UNETS, LANDSAT vs ALL DATA

refer to notebooks in history/experiment_notebooks:

Exp_1_1_UNET_full_9k_Feature32

Exp_1_2_UNET_full_testing_downsample2

Exp_1_3_UNET_full_testing_MS_19_7

```
In [37]: history_MS_landSAT = pd.read_csv("history_MS.csv")
history_Feature32_landSAT = pd.read_csv("history_feature32.csv")
history_downSample2_landSAT = pd.read_csv("history_DownSample2.csv")
```

history_MS_landSAT

Max Epoch F1_val 20 0.667363

history_Feature32_landSAT

Max Epoch F1_val 19 0.673406 last epoch: Epoch F1_val 20 0.670704

history_downSample2_landSAT

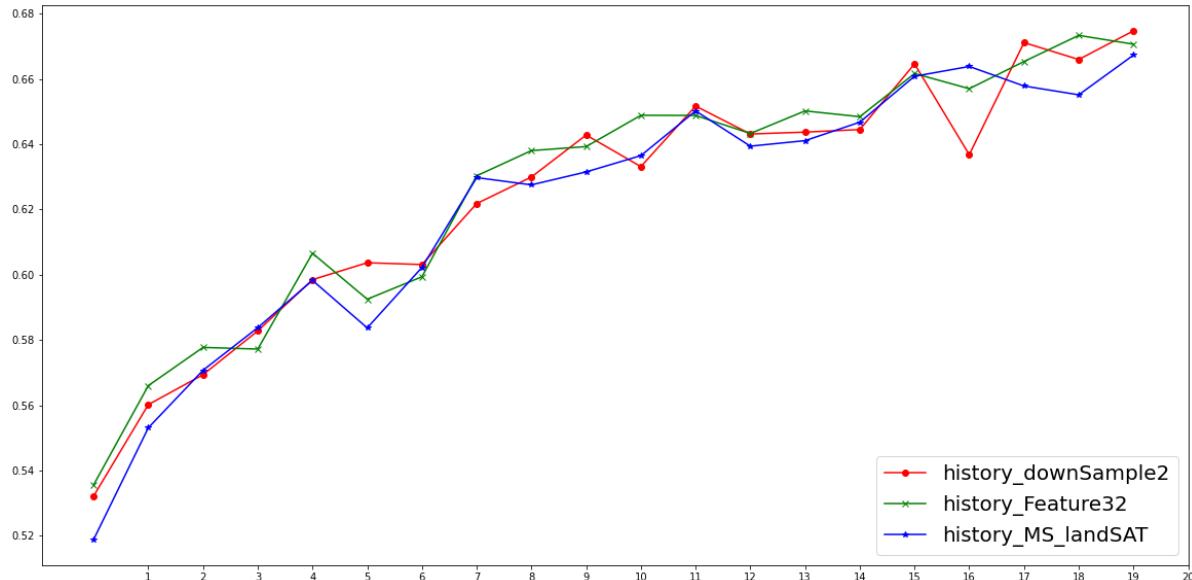
Max Epoch F1_val 20 0.674795

EXP1a-1 Landsat results

```
In [38]: plt.figure(figsize=(20, 10))

plt.plot(history_downSample2_landSAT["F1_val"], 'r', marker='o', label = "history_downSample2_landSAT")
plt.plot(history_Feature32_landSAT["F1_val"], 'g', marker='x', label = "history_Feature32_landSAT")
plt.plot(history_MS_landSAT["F1_val"], 'b', marker='*', label = "history_MS_landSAT")
plt.xticks(range(1,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()
```



EXP1a-2 LandSAT plus Sentinel DataSet

```
In [39]: history_MS_landSAT_full = pd.read_csv("history_MS_19_7.csv")
history_Feature32_full = pd.read_csv("history_feature32_19_7.csv")
history_downSample2_full = pd.read_csv("history_downSample2_19_7.csv")
```

history_MS_landSAT_full

Max Epochs F1_val 16 0.626195 last epoch: Epochs F1_val 20 0.622687

history_Feature32_full

Max Epoch F1_val 19 0.634379 last epoch: Epoch F1_val 20 0.633086

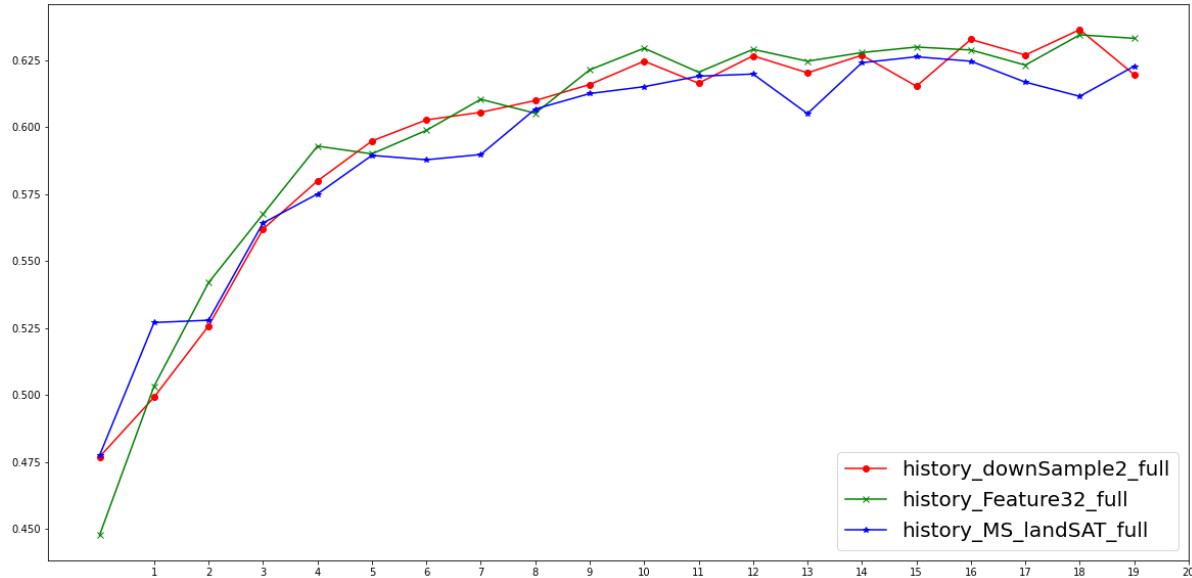
history_downSample2_full

Max Epoch F1_val 19 0.636321 last epoch: Epoch F1_val 20 0.619539

```
In [40]: plt.figure(figsize=(20, 10))

plt.plot(history_downSample2_full["F1_val"], 'r', marker='o', label="history_downSample2")
plt.plot(history_Feature32_full["F1_val"], 'g', marker='x', label="history_Feature32")
plt.plot(history_MS_landSAT_full["F1_val"], 'b', marker='*', label="history_MS_landSAT")
plt.xticks(range(1,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()
```



EXP1b - Histogram Equalization Data on UNET32 Feature

scheduler not used

ADAM used, lr = 0.003

9K images used

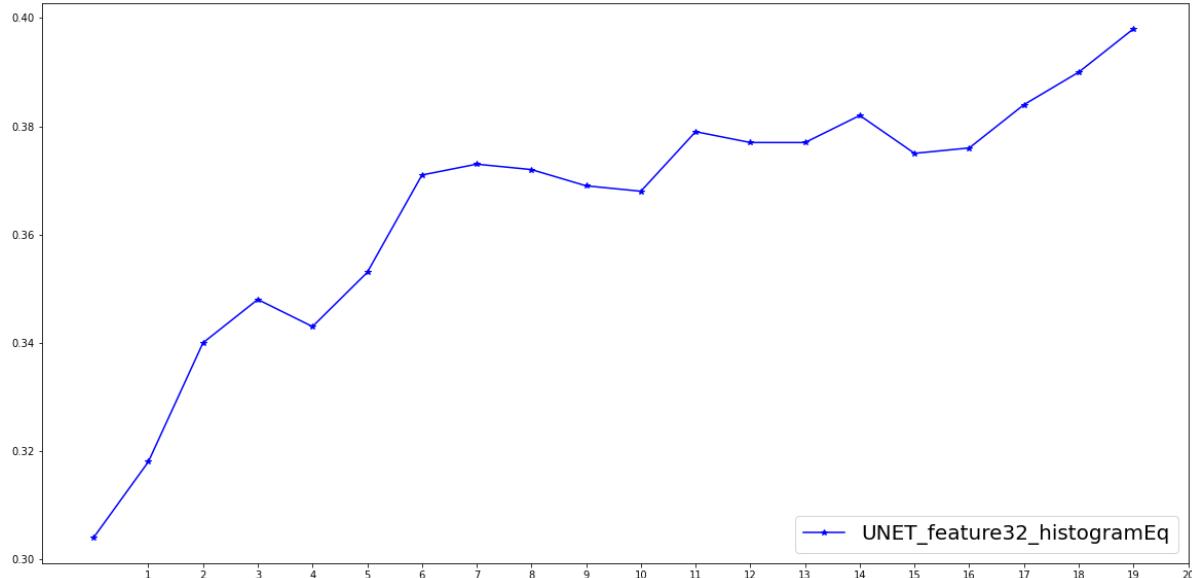
refer to notebooks in history/experiment_notebooks:

Exp_1b_UNET_full_Feature32_HistogramEqualization

```
In [41]: UNET_feature32_histogramEq = pd.read_csv("UNET_feature32_contrasted_data.csv")
max UNET_feature32_histogramEq step f1_val 19 0.398
```

```
In [42]: plt.figure(figsize=(20, 10))

plt.plot(UNET_feature32_histogramEq["f1_val"], 'b', marker='*', label="UNET_feature32_HistEq")
plt.xticks(range(1,21))
plt.legend(loc="lower right", prop={'size': 20})
plt.show()
```



EXP1c: UNET32 Feature One Channel

scheduler used, gamma=0.1

ADAM used, lr = 0.003

9K images used

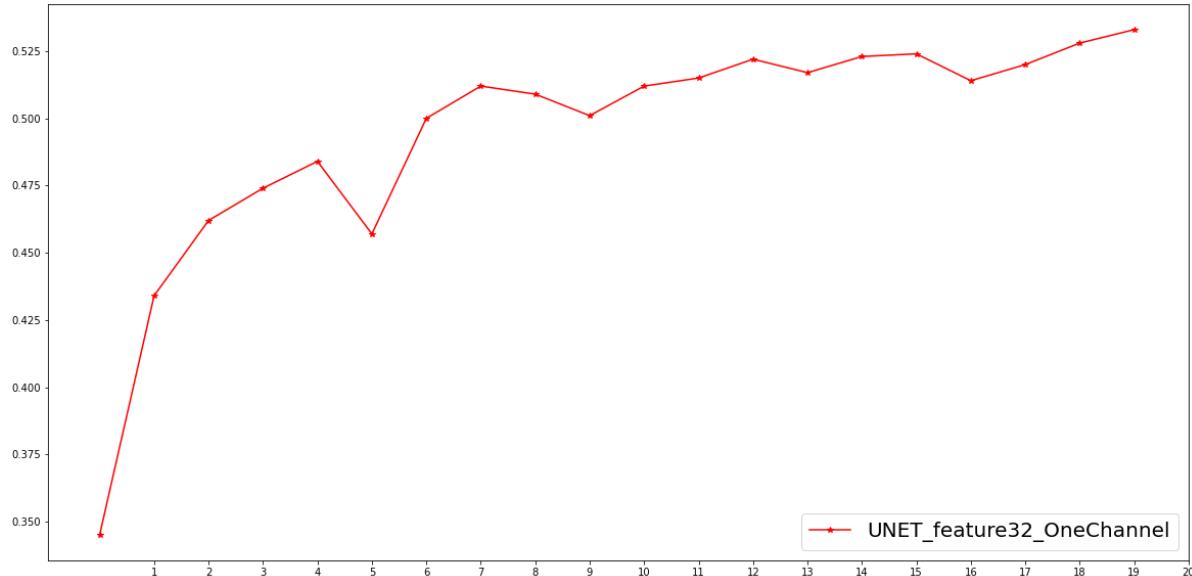
refer to notebooks in history/experiment_notebooks:

Exp_1c - UNET_full_testing_Feature32_onechannel

```
In [43]: UNET_feature32_OneChannel = pd.read_csv("UNET_feature32_one_channel_2022-07-  
# UNET_feature32_OneChannel
```

max UNET_feature32_OneChannel step f1_val 19 0.533

```
In [44]: plt.figure(figsize=(20, 10))  
  
plt.plot(UNET_feature32_OneChannel["f1_val"], 'r', marker='*', label="UNET_feat  
plt.xticks(range(1,21))  
plt.legend(loc="lower right", prop={'size': 20})  
  
plt.show()
```



EXP1e: UNET32 Sen plus LandSAT nADAM vs ADAM

scheduler used, gamma=0.1

ADAM used, lr = 0.003

9K images used

refer to notebooks in history/experiment_notebooks:

Exp_1e_UNET_full_Feature32_NADAM

```
In [45]: UNET_feature32_nAdam = pd.read_csv("FullResults_NADAM.csv")
history_Feature32_full = pd.read_csv("history_feature32_19_7.csv")
```

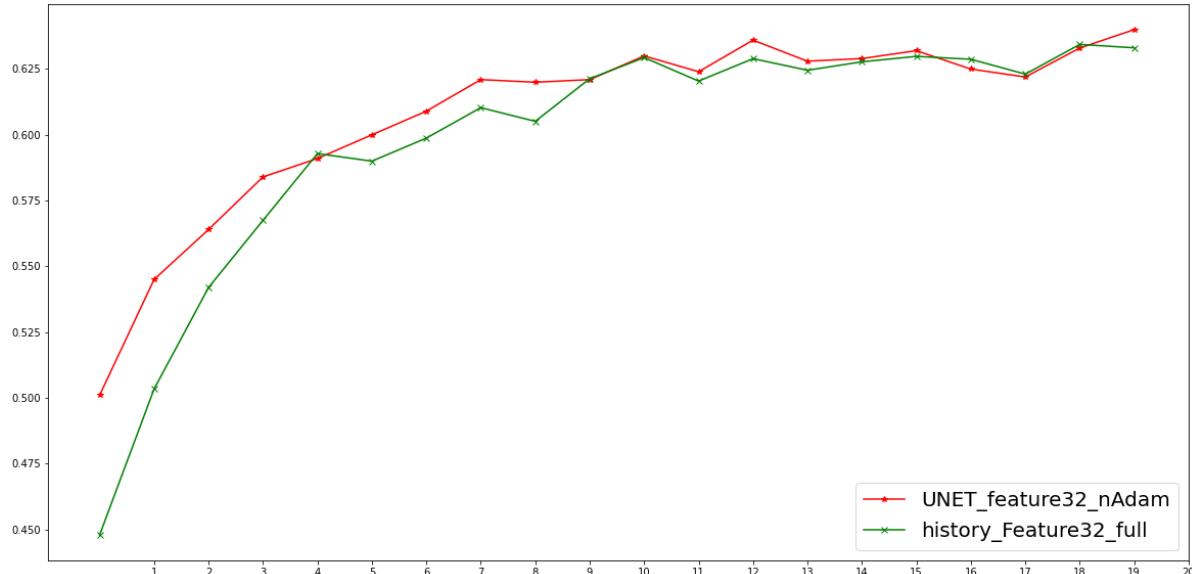
max UNET32 NADAM step f1_val 19 0.64

```
In [46]: plt.figure(figsize=(20, 10))

plt.plot(UNET_feature32_nAdam["f1_val"], 'r', marker='*', label="UNET_feature32_nAdam")
plt.plot(history_Feature32_full["F1_val"], 'g', marker='x', label="history_Feature32_full")

plt.xticks(range(1,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()
```



EXP2a HIL all Data: batch =1, epoch =1

scheduler used, gamma=0.1

ADAM used, lr = 0.003

9K images used

refer to notebooks in history/experiment_notebooks:

Exp_2a_HIL_training_9K

Training complete in 252m 51s

```
In [47]: history_exp1_singleBatch = pd.read_csv("history_exp1_singleBatch.csv")
```

```
### EXP2a HIL all Data max Epoch F1_val 6998 6999 0.381166 last epoch: Epoch F1_val 9108  
0.351684
```

In [48]:

```
x = history_exp1_singleBatch["Epoch"]
y = history_exp1_singleBatch["F1_val"]

y_modded = []
x_modded = []

for i in range(0,9018):# too many points, increment by 100
#    print(i)
    y_modded.append(y[i])
    x_modded.append(i)

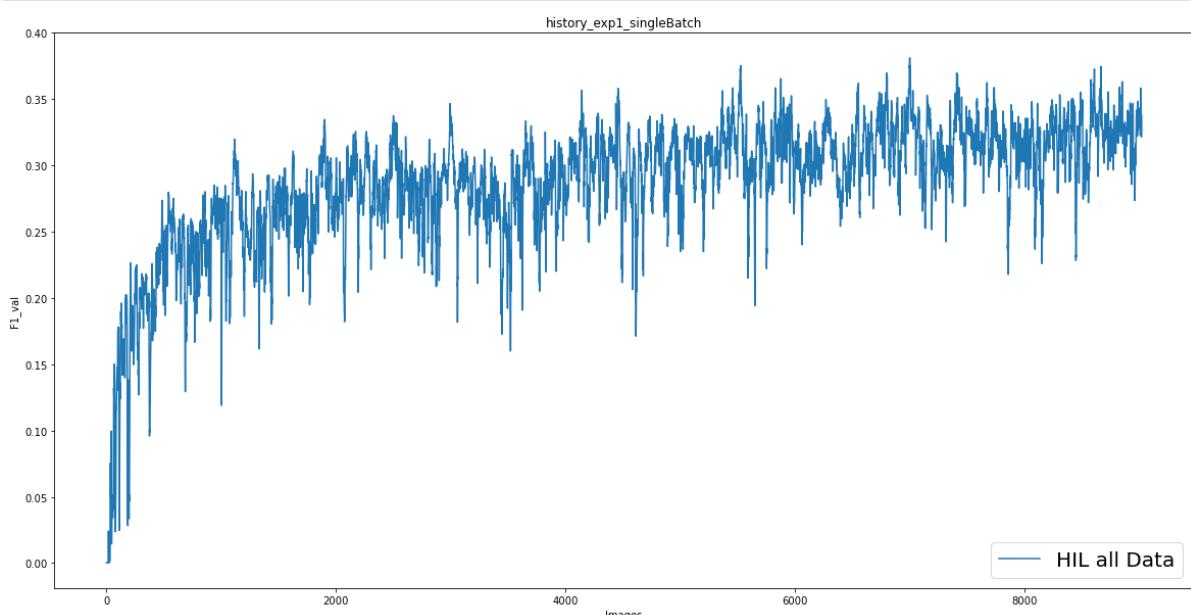
plt.figure(figsize=(20, 10))

plt.plot(x_modded,y_modded,label = "HIL all Data")#, marker='o')

plt.title('history_exp1_singleBatch')
plt.legend(loc="lower right", prop={'size': 20})

plt.xlabel('Images')
plt.ylabel('F1_val')

plt.show()
```



EXP2b HIL on 8K with no BN

scheduler not used

ADAM used, lr = 0.003

8K images used

refer to notebooks in history/experiment_notebooks:

Exp_2b-HIL 8K with no BN

Training complete in 259m 56s

```
In [49]: fileLocationExp2 = os.getcwd() + "//Exp2"
```

```
In [50]: HIL_8K_noBN = pd.read_csv(r"{}//f1_val_Exp2b-HIL_Training8K_no_batch_norm.csv".format(fileLocationExp2))
```

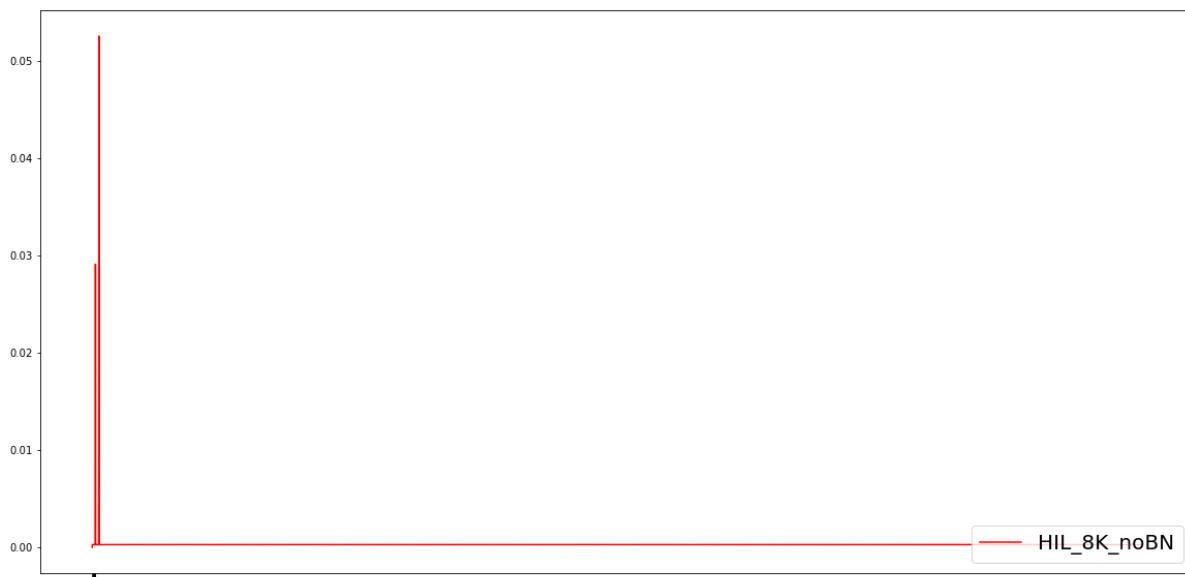
EXP2b HIL on 8K with no BN max step f1_val 52 0.052557 last epoch: step f1_val 8108 0.000286

```
In [51]: plt.figure(figsize=(20, 10))

plt.plot(HIL_8K_noBN["f1_val"], 'r', label="HIL_8K_noBN")

plt.xticks(range(1,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()
```



EXP2c_1- Exponential Moving Average_scheduler_removed_BetaZero9997

scheduler not used

ADAM used, lr = 0.003

9K images used

refer to notebooks in history/experiment_notebooks:

Exp_2c-Exponential Moving Average-Beta0.9997

Training complete in 333m 22s

```
In [52]: ExpMA_zero9997 = pd.read_csv(r"{}/f1_val_2022-08-11_Exp2c-Exponential Moving Average_scheduler_removed_BetaZero9997.ipynb")
```

```
In [53]: maxf1 = ExpMA_zero9997[ExpMA_zero9997["f1_val"]==max(ExpMA_zero9997["f1_val"])]  
print("## EXP2c_1- Exponential Moving Average_scheduler_removed_BetaZero9997")  
  
print(ExpMA_zero9997["f1_val"].tail(1))  
  
maxf1
```

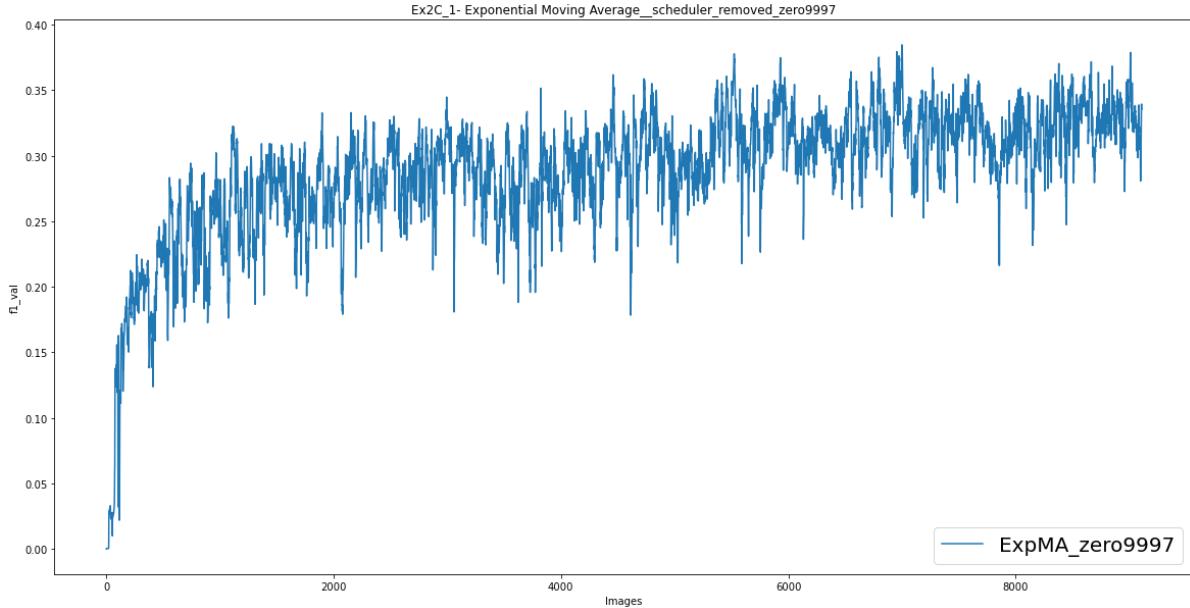
```
## EXP2c_1- Exponential Moving Average_scheduler_removed_BetaZero9997  
9108    0.335406  
Name: f1_val, dtype: float64
```

```
Out[53]:
```

step	f1_val
6999	6999 0.3846

```
In [54]: x = ExpMA_zero9997["step"]  
y = ExpMA_zero9997["f1_val"]
```

```
plt.figure(figsize=(20, 10))  
  
plt.plot(x,y,label = "ExpMA_zero9997" ), marker='o'  
plt.legend(loc="lower right", prop={'size': 20})  
  
plt.title('Exp2c_1- Exponential Moving Average_scheduler_removed_BetaZero9997')  
plt.xlabel('Images')  
plt.ylabel('f1_val')  
plt.show()
```



EXP2c_2- Exponential Moving Average_scheduler_removed_BetaZero

scheduler not used

ADAM used, lr = 0.003

9K images used

refer to notebooks in history/experiment_notebooks:

Exp_2c-Exponential Moving Average-Beta0

Training complete in 333m 38s

```
In [55]: ExpMA_zero= pd.read_csv(r"{}"/f1_val_2022-08-11_Exp2c-Exponential Moving Average_scheduler_removed_BetaZero.csv")  
In [56]: maxf1 = ExpMA_zero[ExpMA_zero["f1_val"]==max(ExpMA_zero["f1_val"])]  
print("## EXP2c_1- Exponential Moving Average_scheduler_removed_BetaZero")  
  
print(ExpMA_zero["f1_val"].tail(1))  
  
maxf1  
  
## EXP2c_1- Exponential Moving Average_scheduler_removed_BetaZero  
9108      0.334122  
Name: f1_val, dtype: float64  
Out[56]:      step      f1_val  
8015    8015    0.374661
```

In [57]:

```
x = ExpMA_zero["step"]
y = ExpMA_zero["f1_val"]

plt.figure(figsize=(20, 10))

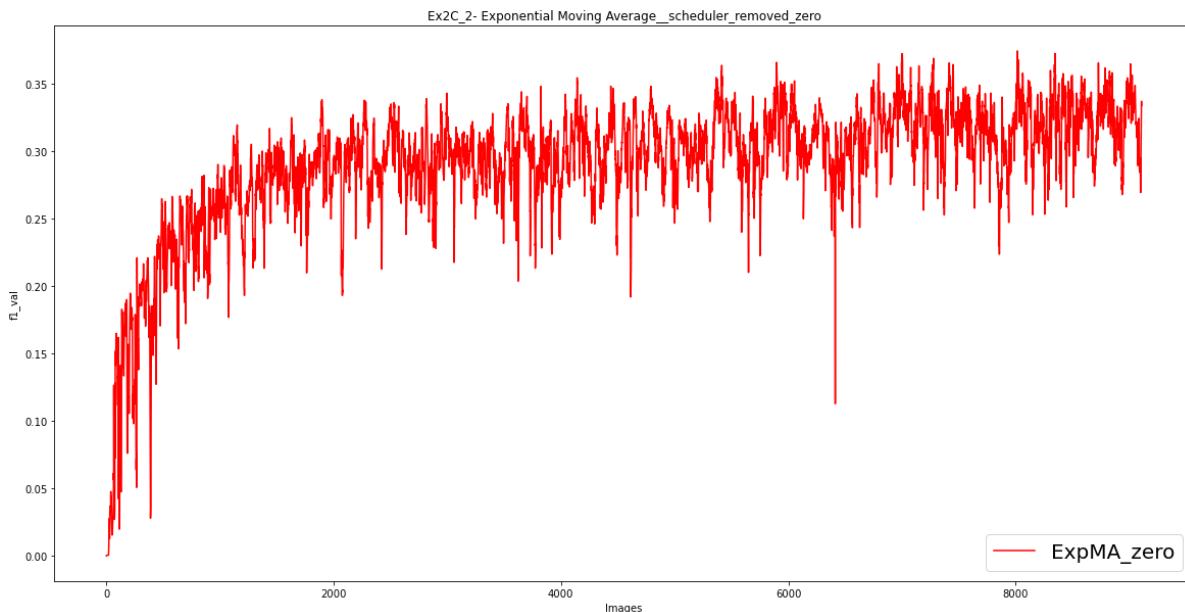
plt.plot(x,y,'r',label = "ExpMA_zero" )#, marker='o')
# plt.plot(x,ExpMA_zero9997["f1_val"],'b',label = "ExpMA_zero9997")#, marker='o')

plt.legend(loc="lower right", prop={'size': 20})

plt.title('Ex2C_2- Exponential Moving Average__scheduler_removed_zero')

plt.xlabel('Images')
plt.ylabel('f1_val')

plt.show()
```



EXP7d_2-_Exponential Moving Average-Beta0.9997_with_pretrained

scheduler not used

ADAM used, lr = 0.003

9K images used

refer to notebooks in history/experiment_notebooks:

Exp_7d_2_Exponential Moving Average-Beta0.9997_with_pretrained.ipynb

Training complete in 295m 46s

```
In [58]: ExpMA_zero9997_pre= pd.read_csv(r"{}"/f1_val_2022-08-12_Exp_2c_X1_Exponential
```

```
In [59]: maxf1 = ExpMA_zero9997_pre[ExpMA_zero9997_pre["f1_val"]==max(ExpMA_zero9997_
```

```
print("## EXP2c_X1- Exp_2c_X1_Exponential Moving Average-Beta0.9997_with_pre
```

```
print(ExpMA_zero9997_pre["f1_val"].tail(1))
```

```
maxf1
```

```
## EXP2c_X1- Exp_2c_X1_Exponential Moving Average-Beta0.9997_with_pretrained
```

```
8108    0.272768
```

```
Name: f1_val, dtype: float64
```

```
Out[59]:
```

step	f1_val
1	0.468377

```
In [60]: x = ExpMA_zero9997_pre["step"]
```

```
y = ExpMA_zero9997_pre["f1_val"]
```

```
plt.figure(figsize=(20, 10))
```

```
plt.plot(x,y,'r',label = "ExpMA_zero9997_pre" ), marker='o')
```

```
# plt.plot(x,ExpMA_zero9997["f1_val"],'b',label = "ExpMA_zero9997")#, marker
```

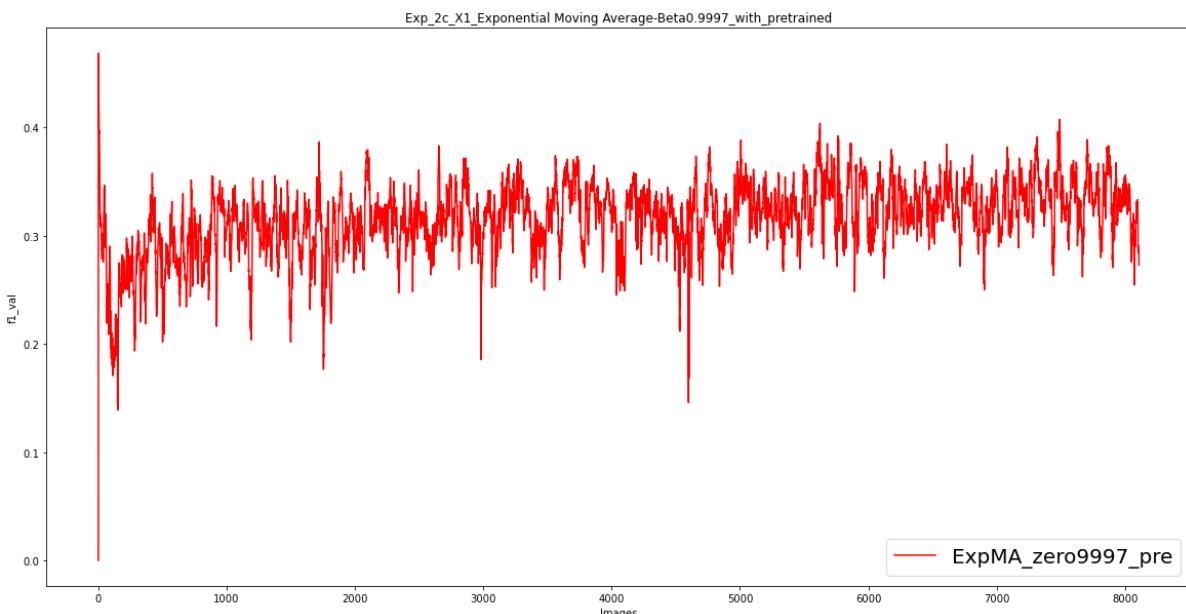
```
plt.legend(loc="lower right", prop={'size': 20})
```

```
plt.title('Exp_2c_X1_Exponential Moving Average-Beta0.9997_with_pretrained')
```

```
plt.xlabel('Images')
```

```
plt.ylabel('f1_val')
```

```
plt.show()
```



EXP3 HIL 10x train each image batch = 1, epoch =1

scheduler used, gamma=0.1

ADAM used, lr = 0.003

9K images used

refer to notebooks in history/experiment_notebooks:

Exp_3_repeat training over the same image

Training complete in 703m 36s

```
In [61]: history_exp4_repeatedImages = pd.read_csv("history_exp4_repeatedImages.csv")
```

```
# EXP3 HIL 10x train each image batch = 1, epoch =1 max: Epoch F1_val 7371 0.3441 last epoch: Epoch F1_val 9108 0.221107
```

```
In [62]: x = history_exp4_repeatedImages ["Epoch"]
y = history_exp4_repeatedImages ["F1_val"]

y_modded = []
x_modded = []

for i in range(0,9018):# too many points, increment by 100
#    print(i)
    y_modded.append(y[i])
    x_modded.append(i)

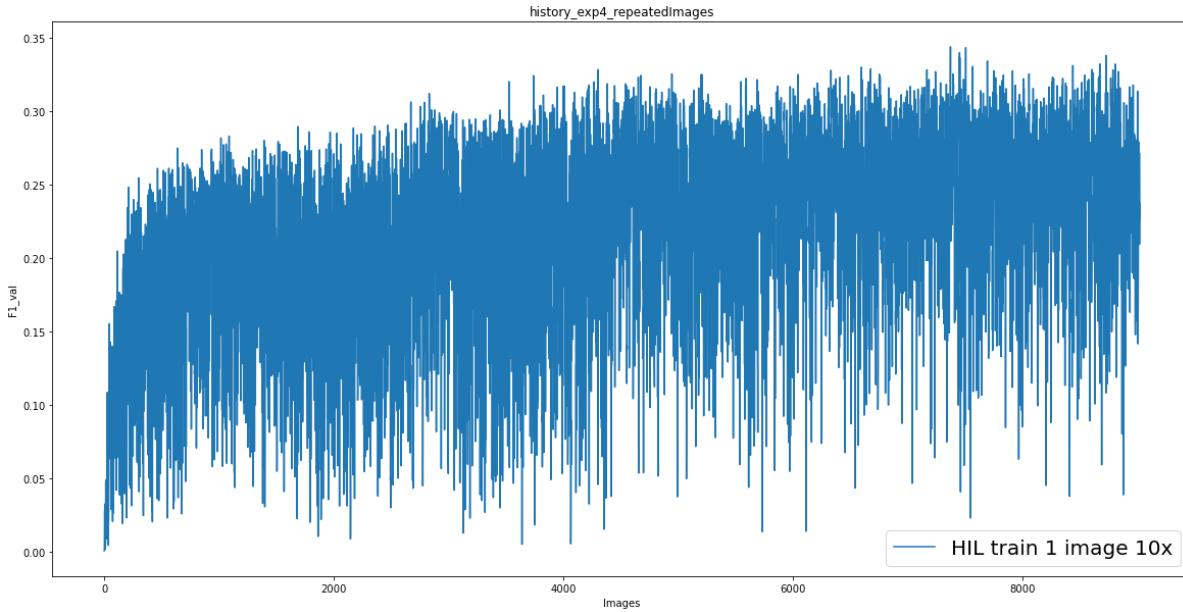
plt.figure(figsize=(20, 10))

plt.plot(x_modded,y_modded,label = "HIL train 1 image 10x")#, marker='o')

plt.title('history_exp4_repeatedImages')
plt.legend(loc="lower right", prop={'size': 20})

plt.xlabel('Images')
plt.ylabel('F1_val')

plt.show()
```



In []:

EXP4: HIL 3X train each UNET, 1 UNET, Batch =1, 1 Epoch

scheduler used, gamma=0.1

ADAM used, lr = 0.003

9K images used

refer to notebooks in history/experiment_notebooks:

Exp_4_Recursion in training

Training complete in 505m 21s

```
In [63]: history_recursion = pd.read_csv("Exp5_recursion_f1.csv")

x = history_recursion["Image"]
y = history_recursion["F1_val"]

y = y.replace('F1 val:d*', '', regex=True).astype (float)

x = x[:9108]
y = y[:9108]
```

EXP4: HIL 3X train each UNET, 1 UNET, Batch =1, 1 Epoch max: 6782 0.264 Name: F1_val, dtype: float64 last epoch: 9107 0.145 Name: F1_val, dtype: float64

```
In [64]: y_modded = []
x_modded = []

for i in range(0,9108):#,,100):# too many points, increment by 100
#    print(i)
    y_modded.append(y[i])
    x_modded.append(i)

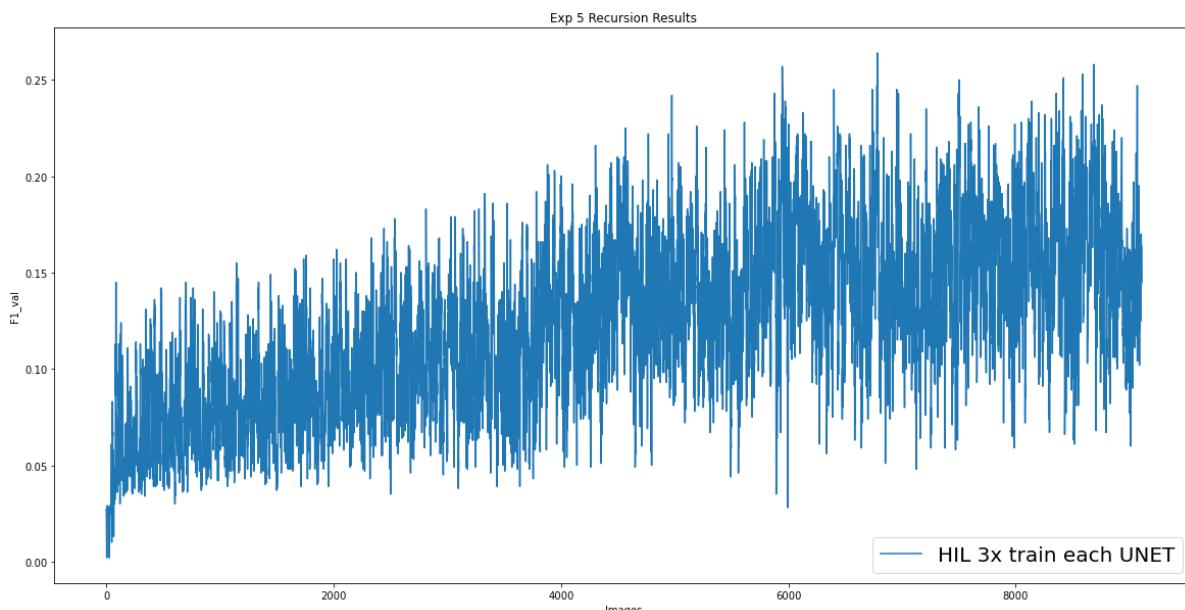
plt.figure(figsize=(20, 10))

plt.plot(x_modded,y_modded, label = "HIL 3x train each UNET") #, marker='o')

plt.title('Exp 5 Recursion Results')
plt.legend(loc="lower right", prop={'size': 20})

plt.xlabel('Images')
plt.ylabel('F1_val')

plt.show()
```



```
In [ ]:
```

EXP5 HIL augmented batch: batch = 1+7, 1 epoch

scheduler used, gamma=0.1

ADAM used, lr = 0.003

9K images used

refer to notebooks in history/experiment_notebooks:

Exp_5_Augmentation_HIL

Training complete in 487m 15s

```
# EXP5 HIL augmented batch: batch = 1+7, 1 epoch max: 5951 0.384347 Name: F1_val, dtype: float64
last epoch: 9107 0.25379 Name: F1_val, dtype: float64
```

```
In [65]: history_exp2_augmentation = pd.read_csv("history_exp2_augmentation.csv")
```

```
x = history_exp2_augmentation["Epoch"]
y = history_exp2_augmentation["F1_val"]
```

```
In [66]:
```

```
y_modded = []
x_modded = []

for i in range(0,9018):# too many points, increment by 100
#    print(i)
    y_modded.append(y[i])
    x_modded.append(i)

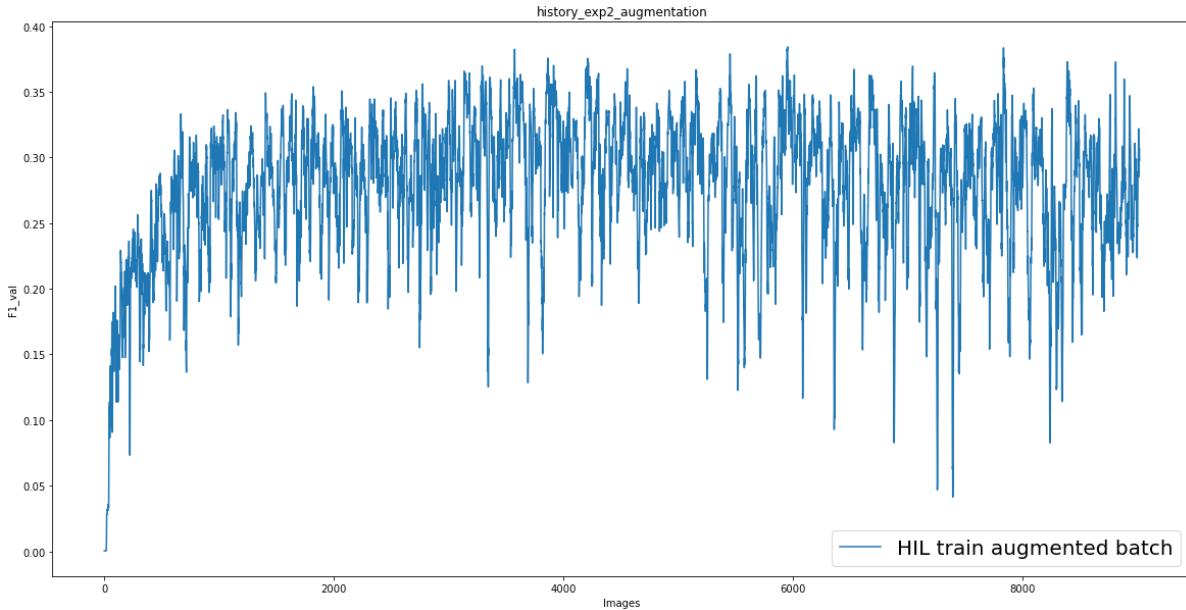
plt.figure(figsize=(20, 10))

plt.plot(x_modded,y_modded,label ="HIL train augmented batch" ), marker='o'
plt.legend(loc="lower right", prop={'size': 20})

plt.title('history_exp2_augmentation')

plt.xlabel('Images')
plt.ylabel('F1_val')

plt.show()
```



EXP 7 PRETRAINED UNET

EXP7a UNET TRAINED ON 1K images

scheduler used, gamma=0.1

ADAM used, lr = 0.003

1K images used

refer to notebooks in history/experiment_notebooks:

Exp_7a_UNET_Feature32_1kTrain

Training complete in 3m 17s

```
In [67]: exp7a = pd.read_csv(r"{}UNET_feature32_1kOnly_FullResult.csv".format(fileLoc))
```

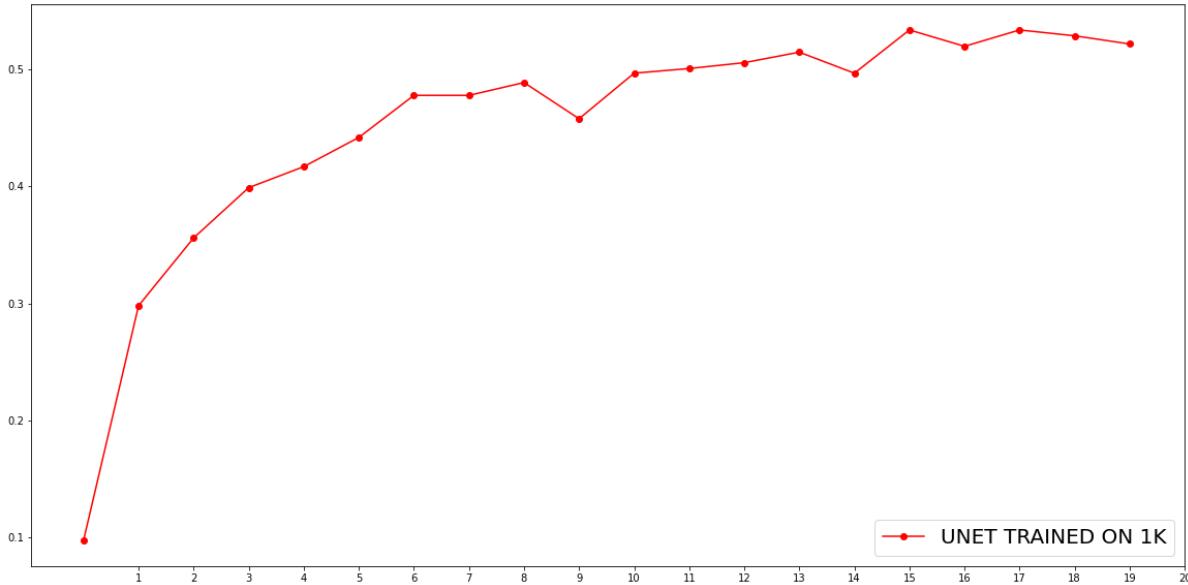
```
## EXP7a UNET TRAINED ON 1K images max: step: f1_val 15 0.534 17 0.534 last epoch: step: f1_val 19 0.522
```

```
In [68]: plt.figure(figsize=(20, 10))

plt.plot(exp7a["f1_val"], 'r', marker='o', label = "UNET TRAINED ON 1K")

plt.xticks(range(1,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()
```



EXP7b HIL training 8k from scratch

scheduler used, gamma=0.1

ADAM used, lr = 0.003

8K images used

refer to notebooks in history/experiment_notebooks:

Exp_7b_HIL_training_8K

Training complete in 224m 36s

```
In [69]: hil_training8k_from_scratch = pd.read_csv(r"{}f1_val_2022-07-20_HIL_Train_8K.csv".format(data_dir))
```

```
## EXP7b HIL training 8k from scratch max: step f1_val 7492 0.387978 last epoch: step f1_val 8108 0.249954
```

```
In [70]: # hil_training8k_from_scratch[750:780]
```

```
In [71]: x = hil_training8k_from_scratch["step"]
y = hil_training8k_from_scratch["f1_val"]

y_modded = []
x_modded = []

for i in range(len(x)):# too many points, increment by 100
#    print(i)
    y_modded.append(y[i])
    x_modded.append(i)

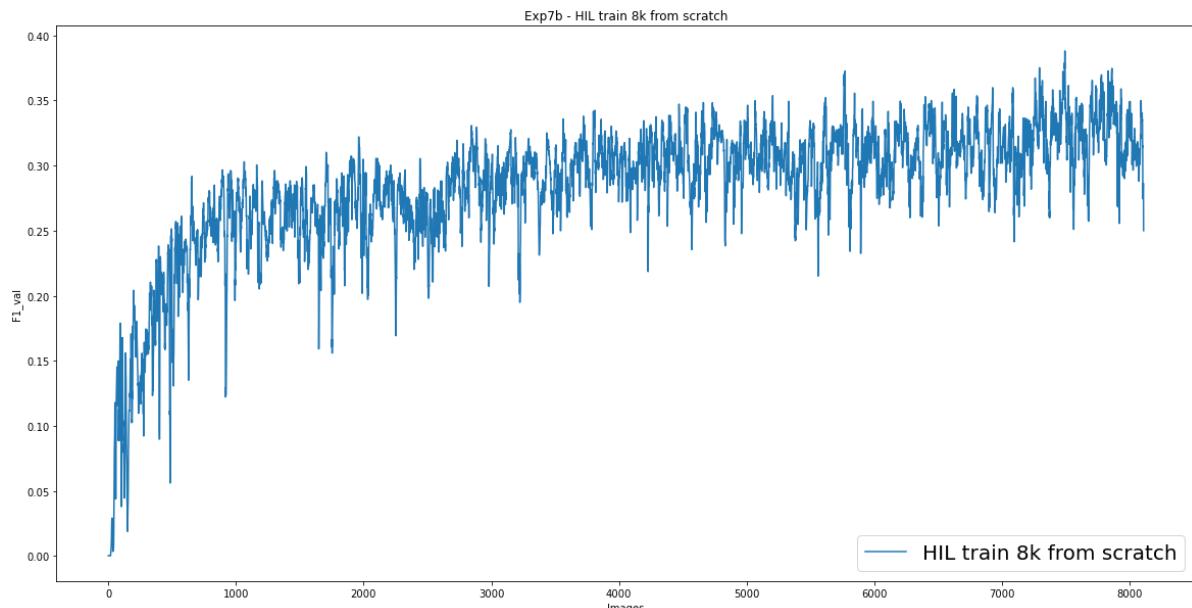
plt.figure(figsize=(20, 10))

plt.plot(x_modded,y_modded,label ="HIL train 8k from scratch" ), marker='o'
plt.legend(loc="lower right", prop={'size': 20})

plt.title('Exp7b - HIL train 8k from scratch')

plt.xlabel('Images')
plt.ylabel('F1_val')

plt.show()
```



Exp 7c - train on fixed 1K and 1 image from 8K from scratch

scheduler used, gamma=0.1

ADAM used, lr = 0.003

fixed 1K and 1 image from 8K

refer to notebooks in history/experiment_notebooks:

Exp_7c train from scratch, 1000 pre-trained set + 1 from 8K, 32 batch, 1 epoch

Training complete in ()

```
In [72]: thousandPlus1_from_scratch = pd.read_csv(r"{}history_thousandPlus1_from_scratch.csv".format(path))
```

Exp 7c - train on fixed 1K and 1 image from 8K from scratch max: step f1_val 60 0.583 92 0.583 last epoch: step f1_val 8107 0.546

```
In [73]: x = thousandPlus1_from_scratch["step"]
y = thousandPlus1_from_scratch["f1_val"]

y_modded = []
x_modded = []

for i in range(len(x)):# too many points, increment by 100
#    print(i)
    y_modded.append(y[i])
    x_modded.append(i)

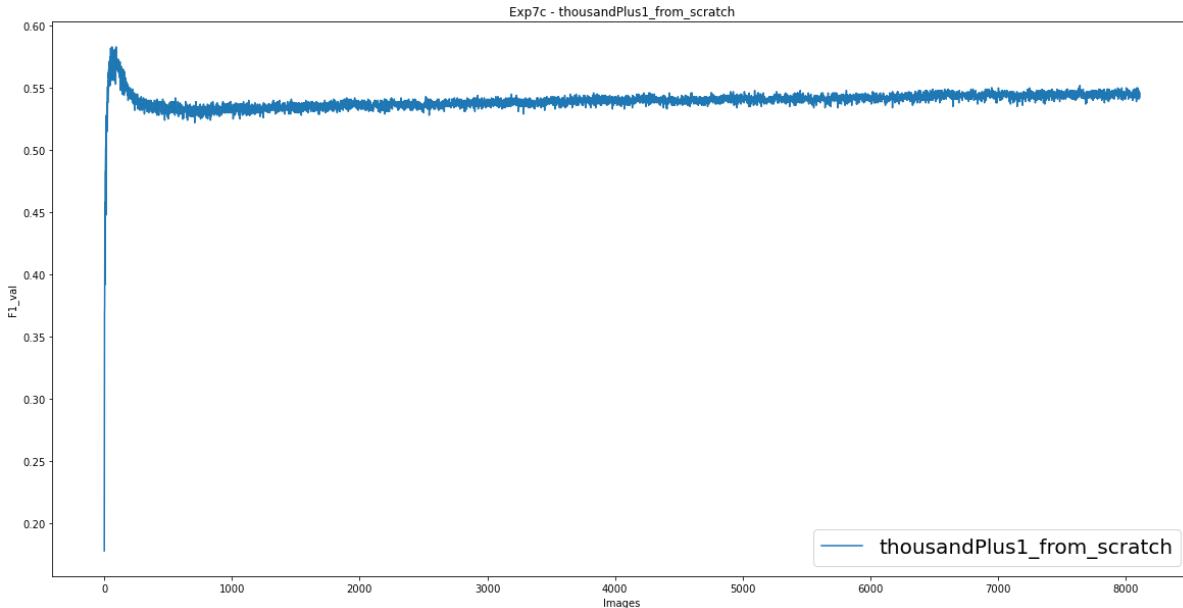
plt.figure(figsize=(20, 10))

plt.plot(x_modded,y_modded,label ="thousandPlus1_from_scratch" ), marker='o'
plt.legend(loc="lower right", prop={'size': 20})

plt.title('Exp7c - thousandPlus1_from_scratch')

plt.xlabel('Images')
plt.ylabel('F1_val')

plt.show()
```



```
In [74]: max(thousandPlus1_from_scratch["f1_val"])
```

```
Out[74]: 0.583
```

7d HIL Training using pretrained model 1k

scheduler used, gamma=0.1

ADAM used, lr = 0.003

8K images used

refer to notebooks in history/experiment_notebooks:

Exp_7d-HIL_train_8k_pretrained

Training complete in 279m 35s

```
In [75]: hil_training8k_from_pretrained = pd.read_csv(r"{}f1_val_2022-07-20_pretrain.csv")
```

```
## 7d HIL Training using pretrained model 1k max: step f1_val 1 0.468053 last epoch: step f1_val 8108 0.2858
```

```
In [76]: x = hil_training8k_from_pretrained["step"]
y = hil_training8k_from_pretrained["f1_val"]

y_modded = []
x_modded = []

for i in range(len(x)):# too many points, increment by 100
#    print(i)
    y_modded.append(y[i])
    x_modded.append(i)

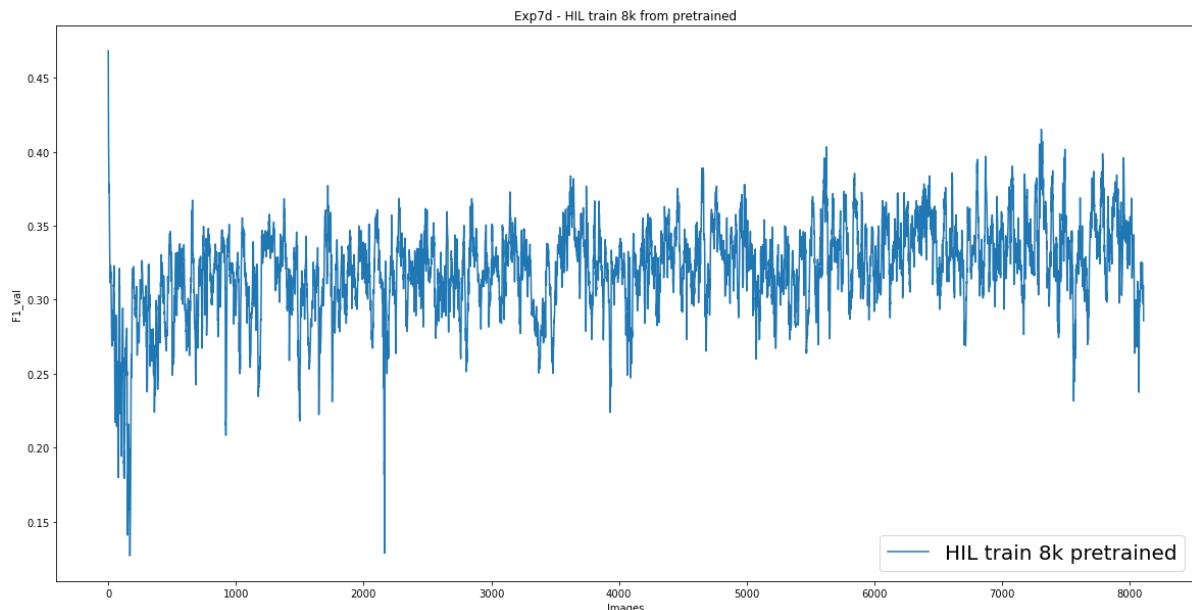
plt.figure(figsize=(20, 10))

plt.plot(x_modded,y_modded,label ="HIL train 8k pretrained" ), marker='o')
plt.legend(loc="lower right", prop={'size': 20})

plt.title('Exp7d - HIL train 8k from pretrained')

plt.xlabel('Images')
plt.ylabel('F1_val')

plt.show()
```



```
In [77]: max(hil_training8k_from_pretrained["f1_val"])
```

```
Out[77]: 0.468053162
```

```
In [78]: max(hil_training8k_from_scratch["f1_val"])
```

```
Out[78]: 0.387977898
```

Exp 7e Start with pretrained UNET

HIL train batch = 1 + b, b = 1, 15, 31

Exp 7e batch = 1 + 1

scheduler used, gamma=0.1

ADAM used, lr = 0.003

fixed 1K and 1 image from 8K

refer to notebooks in history/experiment_notebooks:

Exp_7e_1kPlus1From8K_pretrained_UNET

Training complete in ____

```
In [79]: batch1plus1 = pd.read_csv(r"{}{}/Results_HIL_8Kplus1K_pretrained.csv".format(folder_name, experiment_name))

## Exp 7e batch = 1 + 1 max: step f1_val 47421 0.56 47423 0.56 last epoch: step f1_val 180487 0.489
```

```
In [80]: x = batch1plus1["step"]
y = batch1plus1["f1_val"]

y_modded = []
x_modded = []

for i in range(len(x)):# too many points, increment by 100
#    print(i)
    y_modded.append(y[i])
    x_modded.append(i)

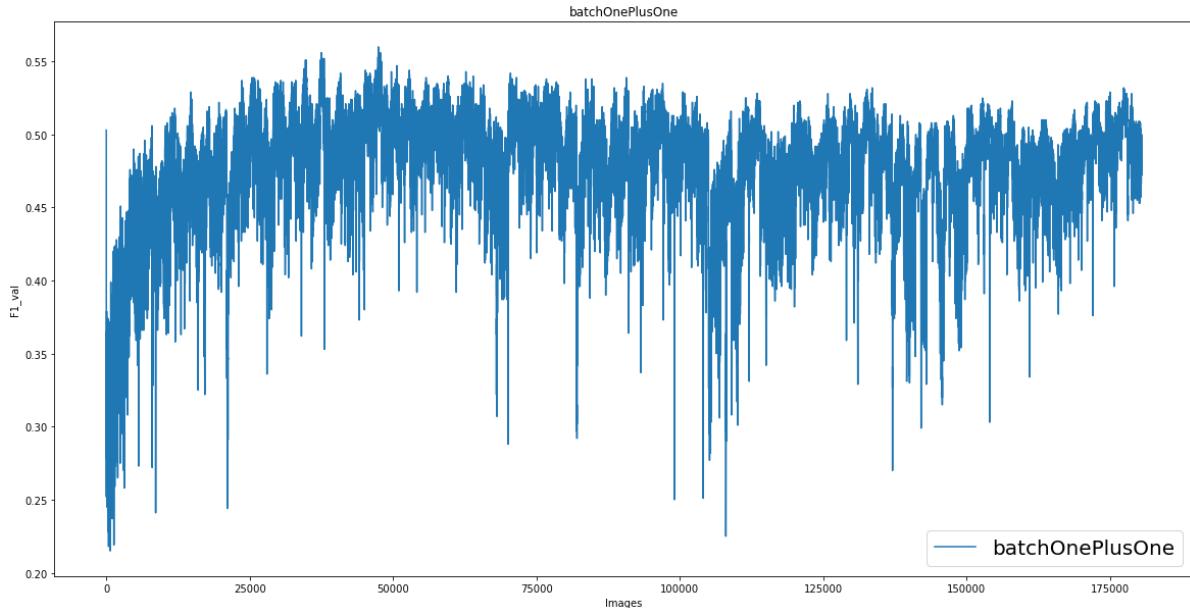
plt.figure(figsize=(20, 10))

plt.plot(x_modded,y_modded,label = "batchOnePlusOne" ), marker='o')
plt.legend(loc="lower right", prop={'size': 20})

plt.title('batchOnePlusOne')

plt.xlabel('Images')
plt.ylabel('F1_val')

plt.show()
```



Exp 7e batch = 1 + 15

scheduler used, gamma=0.1

ADAM used, lr = 0.003

fixed 1K and 1 image from 8K

refer to notebooks in history/experiment_notebooks:

Exp_7e_2_8kPlus1K_Training_16batch-pretrained

Training complete in ____

```
In [81]: batch1plus15 = pd.read_csv(r"{}/Results_HIL_8Kplus1K_pretrained16.csv".format
```

```
## Exp 7e batch = 1 + 15 max: step f1_val 1963 0.589 last epoch: step f1_val 189913 0.533
```

```
In [82]: x = batch1plus15["step"]
y = batch1plus15["f1_val"]

y_modded = []
x_modded = []

for i in range(len(x)):# too many points, increment by 100
#    print(i)
    y_modded.append(y[i])
    x_modded.append(i)

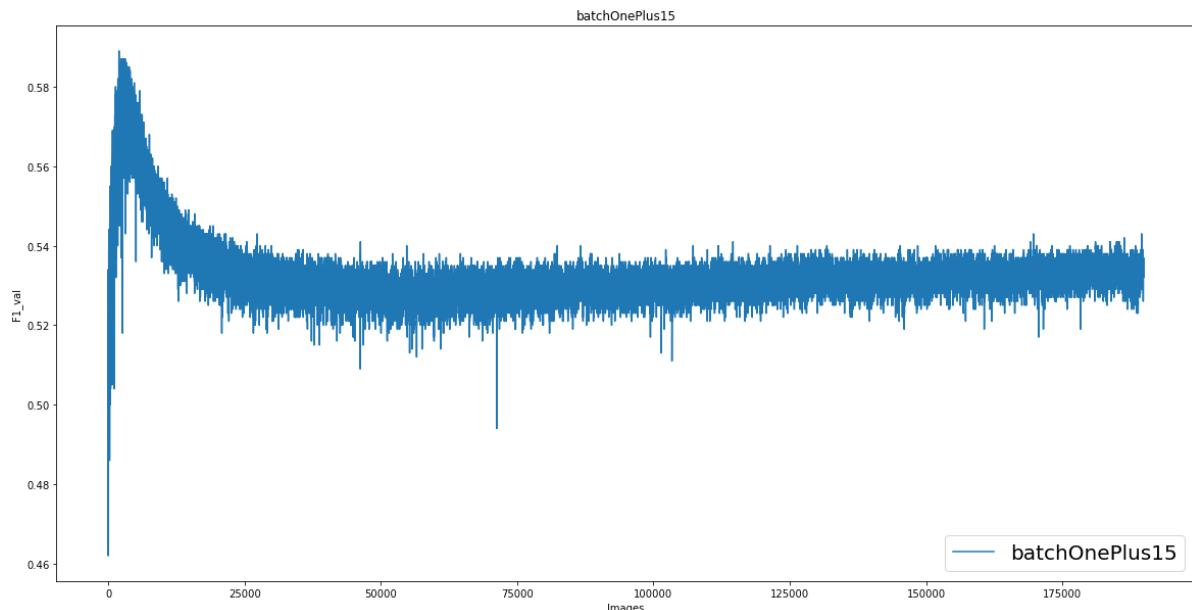
plt.figure(figsize=(20, 10))

plt.plot(x_modded,y_modded,label = "batchOnePlus15" ), marker='o'
plt.legend(loc="lower right", prop={'size': 20})

plt.title('batchOnePlus15')

plt.xlabel('Images')
plt.ylabel('F1_val')

plt.show()
```



Exp 7e batch = 1 + 31

scheduler used, gamma=0.1

ADAM used, lr = 0.003

fixed 1K and 1 image from 8K

refer to notebooks in history/experiment_notebooks:

Exp_7e_3_8kPlus1K_Training_32batch-Pretrained

Training complete in ____

```
In [83]: batch1plus31 = pd.read_csv(r"{}/Results_8Kplus1K_32batch_pretrained.csv".format
```

Exp 7e batch = 1 + 31

max: step f1_val 1066 0.59

last epoch: step f1_val 154671 0.537

```
In [86]: x = batch1plus31["step"]
y = batch1plus31["f1_val"]

y_modded = []
x_modded = []

for i in range(len(x)):# too many points, increment by 100
#    print(i)
    y_modded.append(y[i])
    x_modded.append(i)

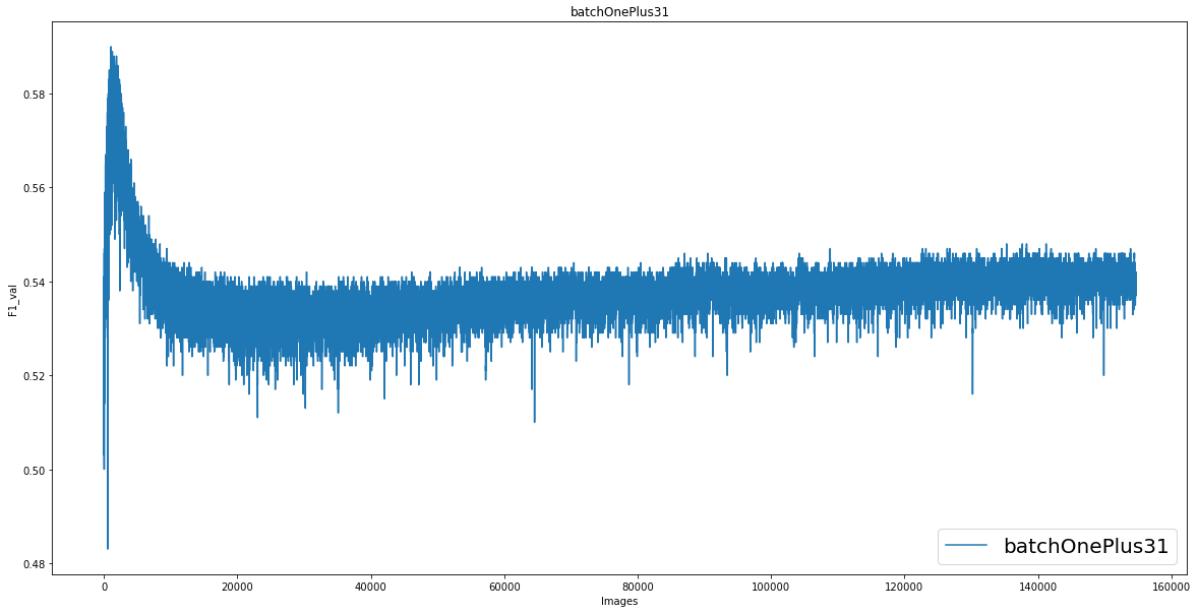
plt.figure(figsize=(20, 10))

plt.plot(x_modded,y_modded,label = "batchOnePlus31" ), marker='o'
plt.legend(loc="lower right", prop={'size': 20})

plt.title('batchOnePlus31')

plt.xlabel('Images')
plt.ylabel('F1_val')

plt.show()
```



```
In [87]: max(batch1plus31["f1_val"])
```

```
Out[87]: 0.59
```

```
In [88]: max(batch1plus1["f1_val"])
```

```
Out[88]: 0.56
```

```
In [89]: max(batch1plus15["f1_val"])
```

```
Out[89]: 0.589
```

Exp 7g Batch 32, 1 epoch. Apply random augmentation to the 1K(batch 31) plus 1 image from 8K(no aug) aug probability = $1/2 * 1/3 = 1/6$

scheduler not used

ADAM used, lr = 0.003

fixed 1K and 1 image from 8K

refer to notebooks in history/experiment_notebooks:

Exp_7g - Apply random aug to 1K + 1image from 8K(no aug)_scheduler_removed

Training complete in 1447m 34s

```
In [90]: Augmented1K_7g = pd.read_csv(r"{}/Results_7g.csv".format(fileLocationExp7))
```

Exp 7g Batch 32, 1 epoch. Apply random augmentation to the 1K(batch 31) plus 1 image from 8K(no aug) aug probability = $1/2 * 1/3 = 1/6$

max: step f1_val 151 0.621

last epoch: step f1_val 8107 0.575

In [93]:

```
x = Augmented1K_7g["step"]
y = Augmented1K_7g["f1_val"]

plt.figure(figsize=(20, 10))

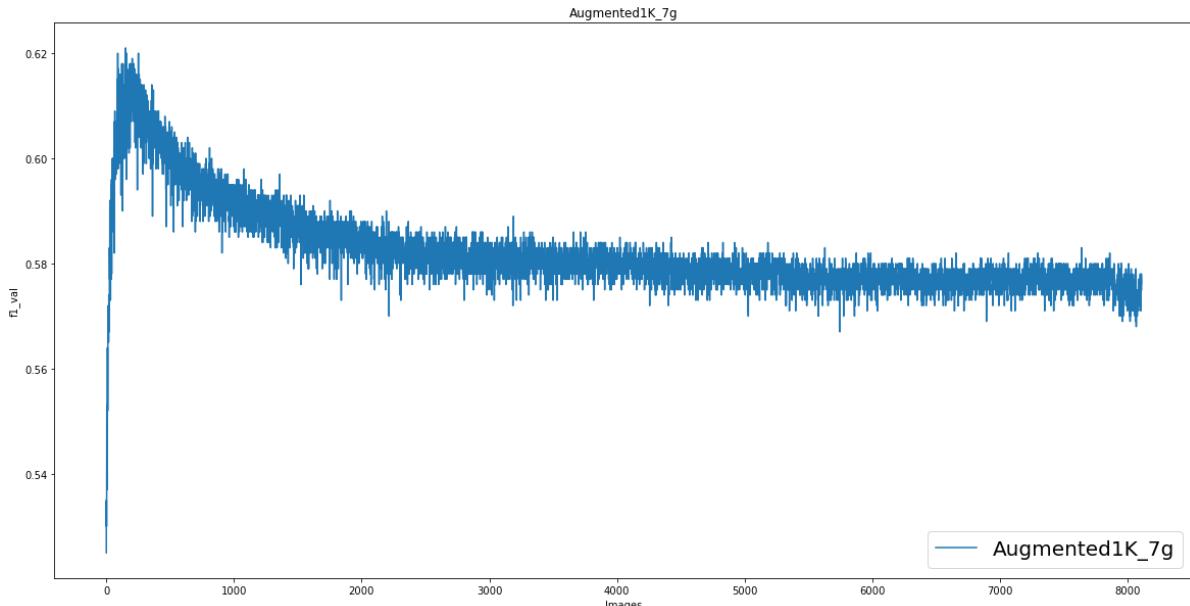
plt.plot(x,y,label = "Augmented1K_7g" ), marker='o'
plt.legend(loc="lower right", prop={"size": 20})

plt.title('Augmented1K_7g')

plt.xlabel('Images')
plt.ylabel('f1_val')

plt.show()

max(y)
```



Out[93]: 0.621

Exp 7h: Add 1 image from 8K to the 1K dataset and retrain using only 512 images randomly chosen.

scheduler not used

ADAM used, lr = 0.003

fixed 1K and 1 image from 8K, only 512 images each iteration

refer to notebooks in history/experiment_notebooks:

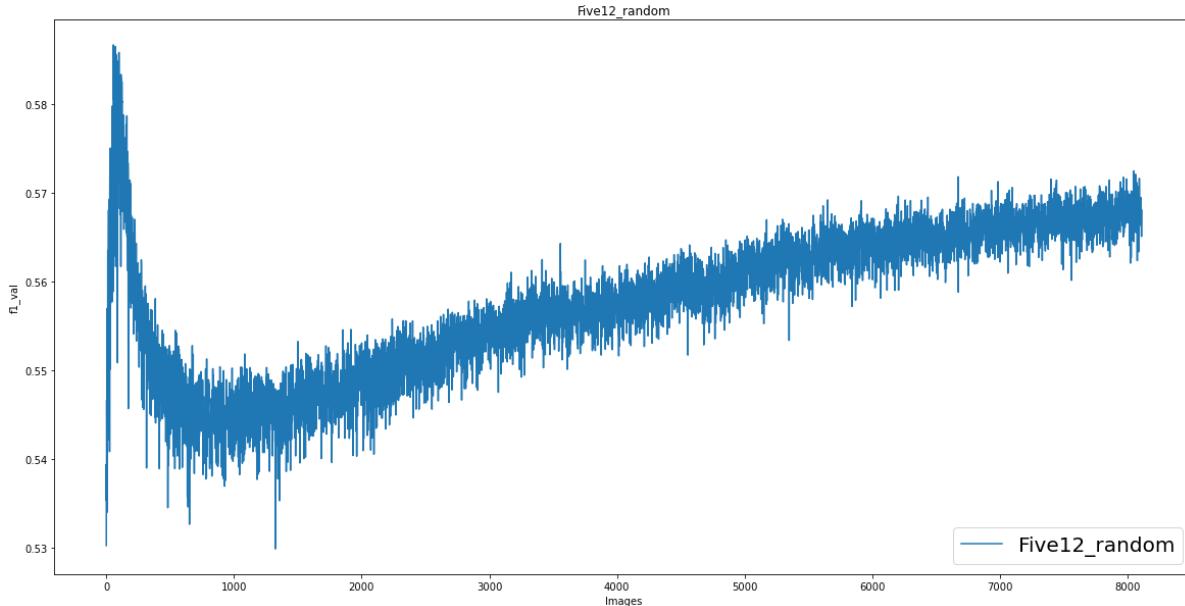
Exp_7h - Add 1 image from 8K to 1K and retrain using only 512 images randomly_scheduler_removed

Training complete in 968m 31s

```
## Exp 7h: Add 1 image from 8K to the 1K dataset and retrain using only 512 images randomly chosen.  
max: step f1_val 57 0.586676 last epoch: step f1_val 8108 0.568022
```

```
In [94]: Five12_random = pd.read_csv(r"{}f1_val_2022-07-26_fiveHundredAndOne_Trainin".format(path))
```

```
In [95]: x = Five12_random["step"]  
y = Five12_random["f1_val"]  
  
plt.figure(figsize=(20, 10))  
  
plt.plot(x,y,label = "Five12_random" ), marker='o'  
plt.legend(loc="lower right", prop={'size': 20})  
  
plt.title('Five12_random')  
  
plt.xlabel('Images')  
plt.ylabel('f1_val')  
  
plt.show()
```



```
In [96]: fileLocationExp9 = os.getcwd() + "//Exp9"
```

Exp9a-HIL_Training8K_SigmoidLayerOnly_val Scheduler not Used

scheduler not used

ADAM used, lr = 0.003

8k images used

Sigmoid layer trained only

refer to notebooks in history/experiment_notebooks:

Exp_9a

Training complete in 270m 22s

```
In [97]: sigmoidLayerOnly = pd.read_csv(r"{}//HIL_Training8K_sigmoidTrainingOnly_val.csv")
```

```
## Exp9a-HIL_Training8K_SigmoidLayerOnly_val Scheduler not Used max: step f1_val 5 0.51863 last epoch: step f1_val 8108 0.443518
```

```
In [98]: x = sigmoidLayerOnly["step"]
y = sigmoidLayerOnly["f1_val"]

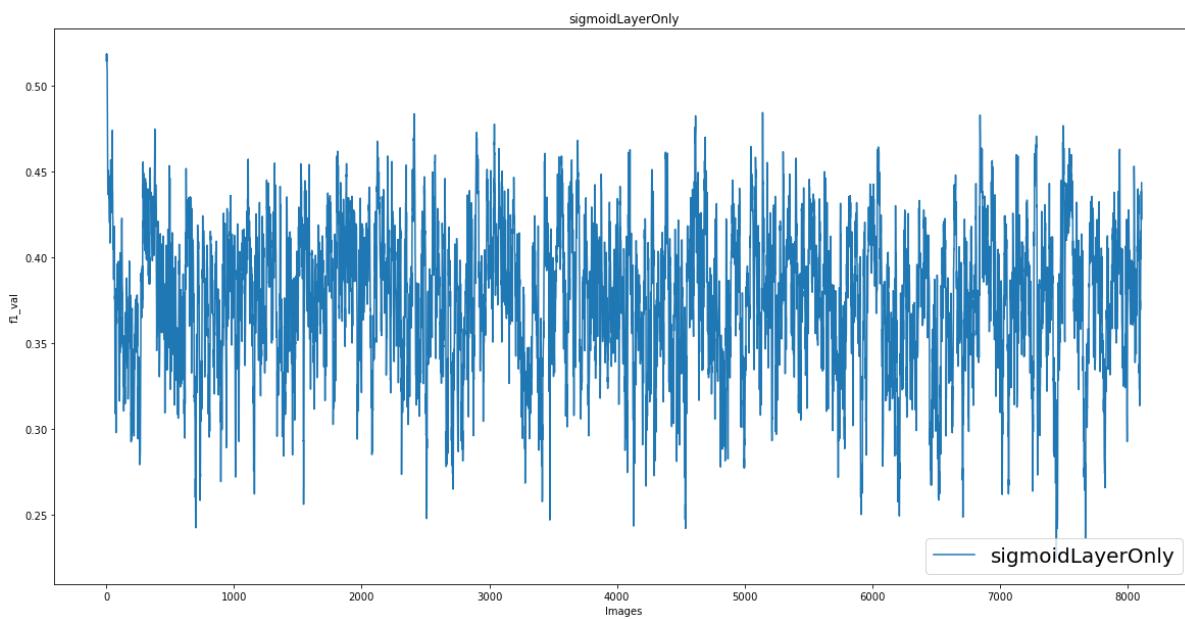
plt.figure(figsize=(20, 10))

plt.plot(x,y,label = "sigmoidLayerOnly" )#, marker='o')
plt.legend(loc="lower right", prop={'size': 20})

plt.title('sigmoidLayerOnly')

plt.xlabel('Images')
plt.ylabel('f1_val')

plt.show()
```



Exp9b-HIL_Training8K_LastLayerTrainingOnly_val Scheduler not Used

scheduler not used

ADAM used, lr = 0.003

8k images used

last layer trained only

refer to notebooks in history/experiment_notebooks:

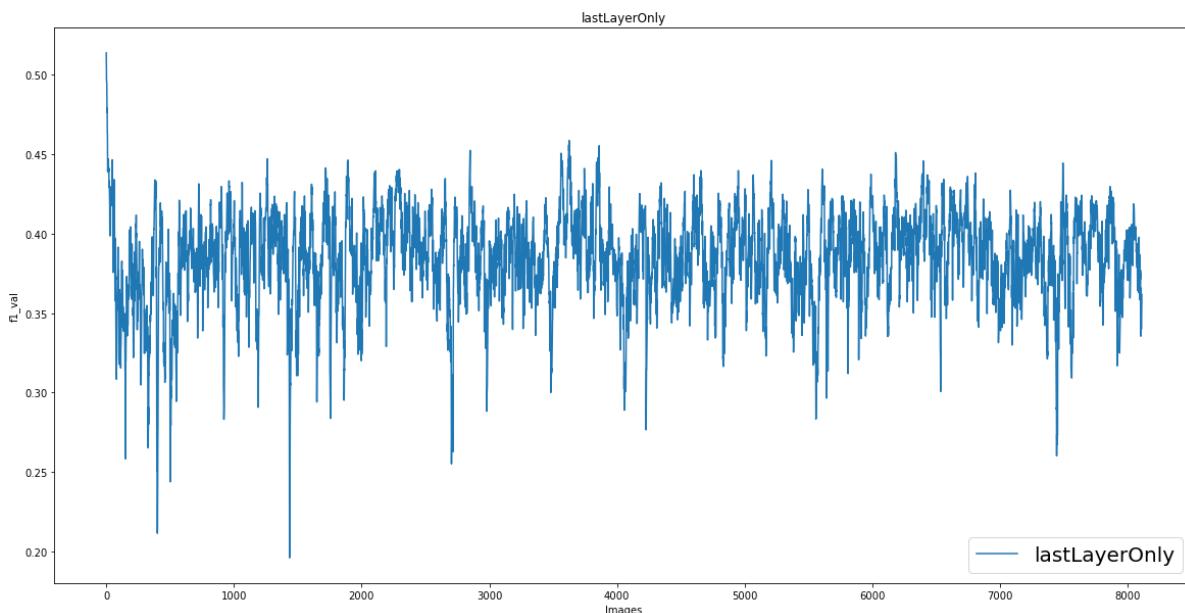
Exp_9b

Training complete in 270m 13s

```
In [99]: lastLayerOnly = pd.read_csv(r"{}/Exp9b-HIL_Training8K_LastLayerTrainingOnly_
```

```
## Exp9b-HIL_Training8K_LastLayerTrainingOnly_val Scheduler not Used max: step f1_val 1 0.513305  
last epoch: step f1_val 8108 0.340326
```

```
In [100... x = lastLayerOnly["step"]  
y = lastLayerOnly["f1_val"]  
  
plt.figure(figsize=(20, 10))  
  
plt.plot(x,y,label = "lastLayerOnly" ), marker='o'  
plt.legend(loc="lower right", prop={'size': 20})  
  
plt.title('lastLayerOnly')  
  
plt.xlabel('Images')  
plt.ylabel('f1_val')  
  
plt.show()
```



Exp9c-HIL_Training8K_Last2LayerTrainingOnly_val Scheduler not Used

scheduler not used

ADAM used, lr = 0.003

8k images used

last layer trained only

refer to notebooks in history/experiment_notebooks:

Exp_9c_last2Layers

Training complete in 282m 34s

```
In [101]: last2LayerOnly = pd.read_csv(r"{}/f1_val_2022-08-09_Exp9c_last2Layers.csv".format(
```

```
In [ ]:
```

```
In [102]: maxf1 = last2LayerOnly[last2LayerOnly['f1_val']==max(last2LayerOnly['f1_val'])]
print("Exp9c-HIL_Training8K_Last2LayerTrainingOnly_val Scheduler not Used")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(last2LayerOnly.tail(1))
```

Exp9c-HIL_Training8K_Last2LayerTrainingOnly_val Scheduler not Used

max:

step	f1_val
1	0.513305

last epoch:

step	f1_val
8108	0.340326

```
In [103]: x = last2LayerOnly["step"]
y = last2LayerOnly["f1_val"]

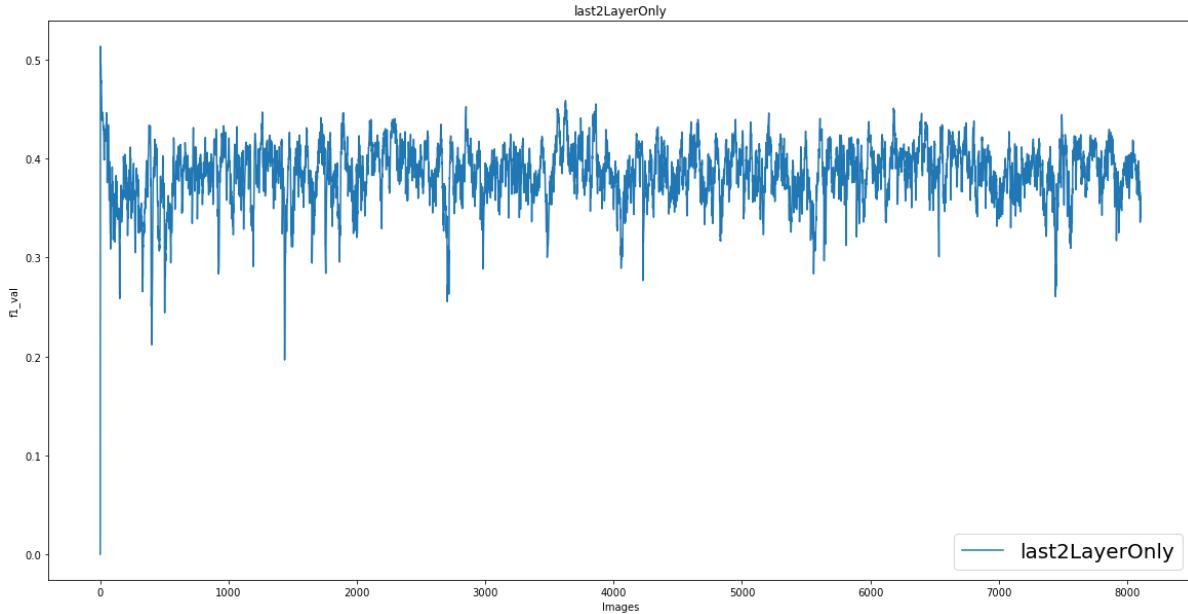
plt.figure(figsize=(20, 10))

plt.plot(x,y,label = "last2LayerOnly" ), marker='o'
plt.legend(loc="lower right", prop={'size': 20})

plt.title('last2LayerOnly')

plt.xlabel('Images')
plt.ylabel('f1_val')

plt.show()
```



Experiment - KmeansLoss

```
In [104...]:  
#### scheduler used  
#### ADAM used, lr = 0.003  
#### 9k images used (all images)  
  
### refer to notebooks in history/experiment_notebooks:  
#### UNET_full_testing_Feature32-KmeansLoss.ipynb  
#### Training complete in 27m 54s  
  
#### Using mean and standard deviation of contour points and non contour poi  
#### to calculate the new loss. The loss was divided by a factor of beta to  
#### not blow up  
#### beta chosen are 5,10,20,50,70
```

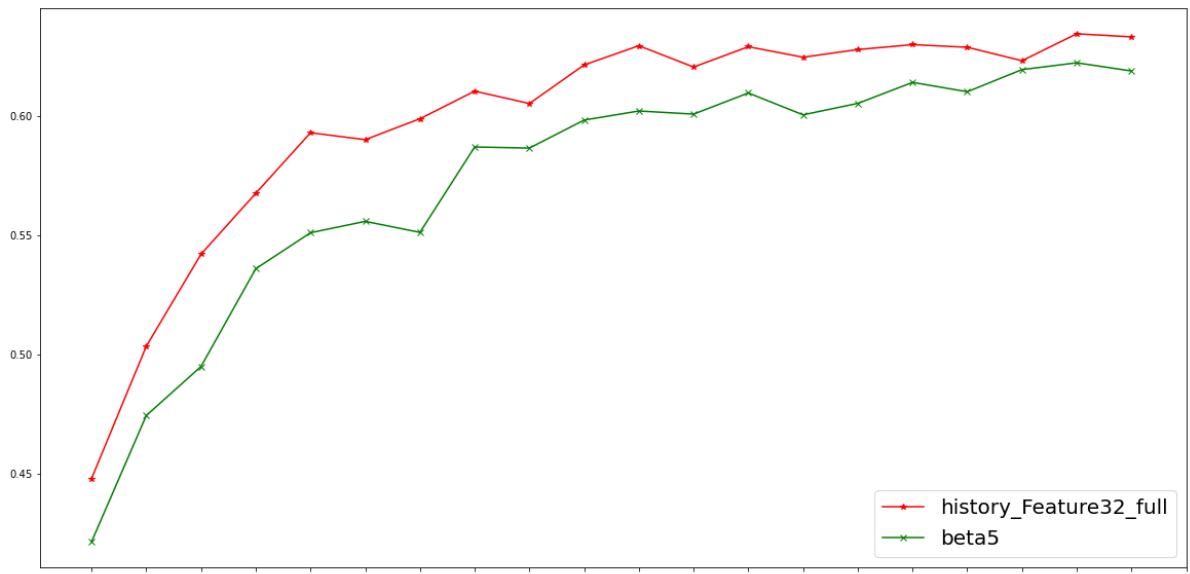
```
In [105...]:  
fileLocationExpKmeans = os.getcwd() + "/KmeansLossExpAndResults"  
history_Feature32_full = pd.read_csv("history_feature32_19_7.csv")  
  
beta5 = pd.read_csv(fileLocationExpKmeans+ "//beta5//f1_val_2022-08-25_Exp_K  
beta10 = pd.read_csv(fileLocationExpKmeans+ "//beta10//f1_val_2022-08-25_Exp_K  
beta20 = pd.read_csv(fileLocationExpKmeans+ "//beta20//f1_val_2022-08-26_Exp_K  
beta50 = pd.read_csv(fileLocationExpKmeans+ "//beta50//f1_val_2022-08-26_Exp_K  
beta70 = pd.read_csv(fileLocationExpKmeans+ "//beta70//f1_val_2022-08-26_Exp_K
```

```
In [106...]:  
maxf1 = history_Feature32_full[history_Feature32_full['F1_val']==max(history_F  
print("history_Feature32_full")  
print("max:")  
print(maxf1)  
print()  
print("last epoch:")  
print(history_Feature32_full.tail(1))
```

```
history_Feature32_full  
max:  
    Epoch      F1_val  
18      19  0.634379  
  
last epoch:  
    Epoch      F1_val  
19      20  0.633086
```

UNET original vs beta5KMeans

```
In [107]: plt.figure(figsize=(20, 10))  
  
plt.plot(history_Feature32_full["F1_val"], 'r', marker='*', label="history_Feat  
plt.plot(beta5["f1_val"], 'g', marker='x', label="beta5")  
  
plt.xticks(range(0,21))  
plt.legend(loc="lower right", prop={'size': 20})  
  
plt.show()  
  
maxf1 = beta5[beta5['f1_val']==max(beta5['f1_val'])]  
print("beta5")  
print("max:")  
print(maxf1)  
print()  
print("last epoch:")  
print(beta5.tail(1))
```



```
beta5  
max:  
    step      f1_val  
18      19  0.622215  
  
last epoch:  
    step      f1_val  
19      20  0.618782
```

UNET original vs beta10KMeans

In [108...]

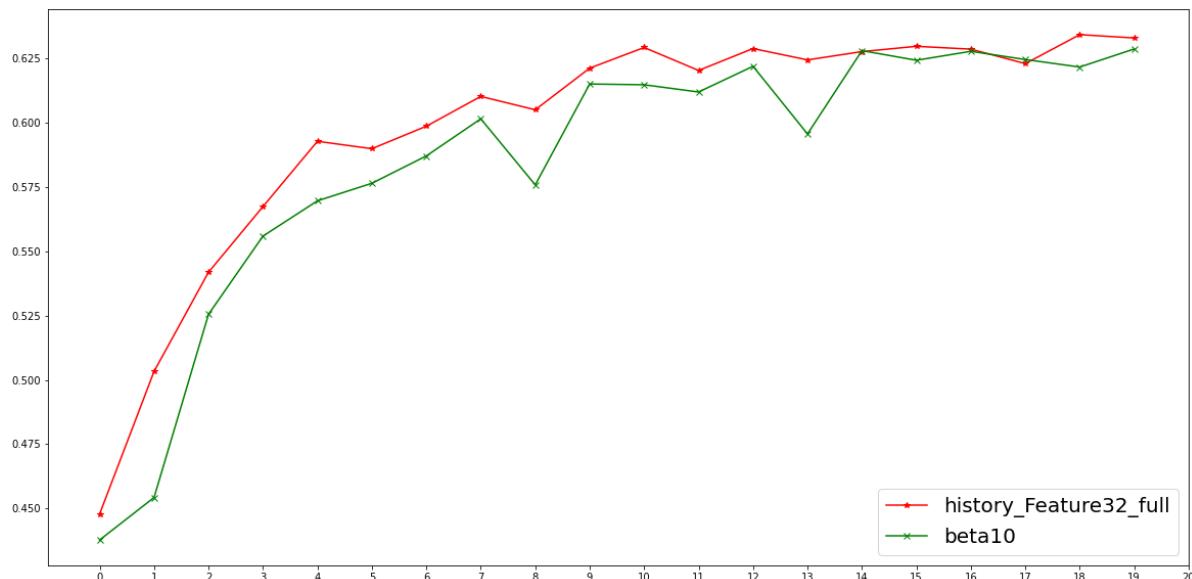
```
plt.figure(figsize=(20, 10))

plt.plot(history_Feature32_full["F1_val"], 'r', marker='*', label="history_Feature32_full")
plt.plot(beta10["f1_val"], 'g', marker='x', label="beta10")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()

maxf1 = beta10[beta10['f1_val']==max(beta10['f1_val'])]
print("beta5")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(beta10.tail(1))
```



beta5

max:

step	f1_val	
19	20	0.628792

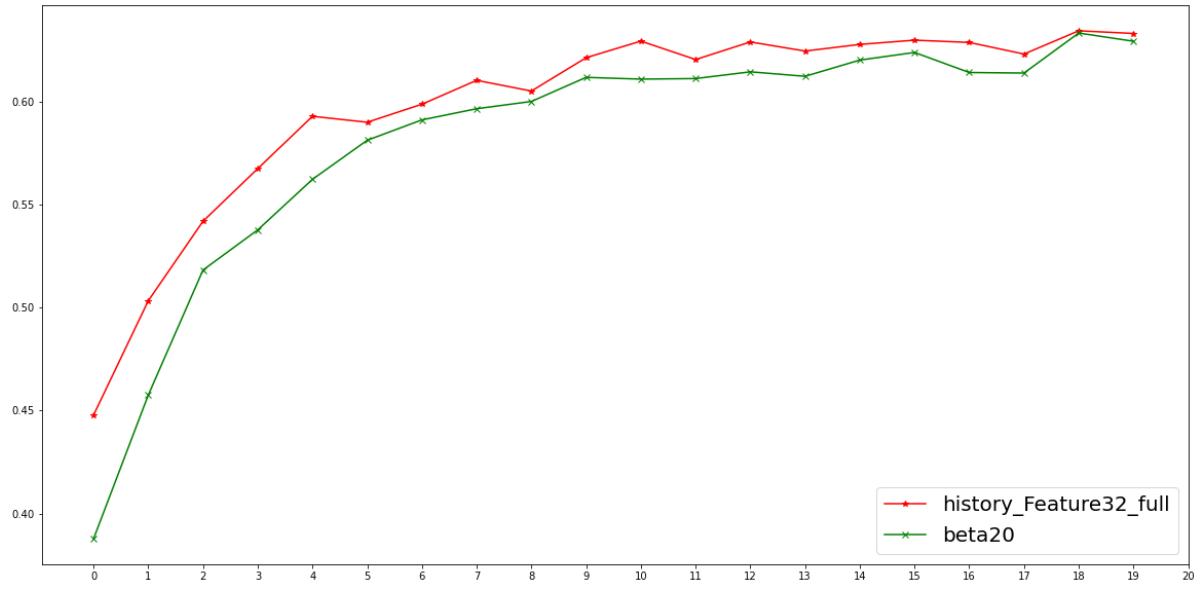
last epoch:

step	f1_val	
19	20	0.628792

UNET original vs beta20KMeans

```
In [109]: plt.figure(figsize=(20, 10))
```

```
plt.plot(history_Feature32_full["F1_val"], 'r', marker='*', label="history_Feat  
plt.plot(beta20["f1_val"], 'g', marker='x', label="beta20")  
  
plt.xticks(range(0,21))  
plt.legend(loc="lower right", prop={"size": 20})  
  
plt.show()  
  
maxf1 = beta20[beta20['f1_val']==max(beta20['f1_val'])]  
print("beta5")  
print("max:")  
print(maxf1)  
print()  
print("last epoch:")  
print(beta20.tail(1))
```



```
beta5  
max:  
    step      f1_val  
18     19   0.633301
```

```
last epoch:  
    step      f1_val  
19     20   0.6293
```

```
In [ ]:
```

UNET original vs beta50KMeans

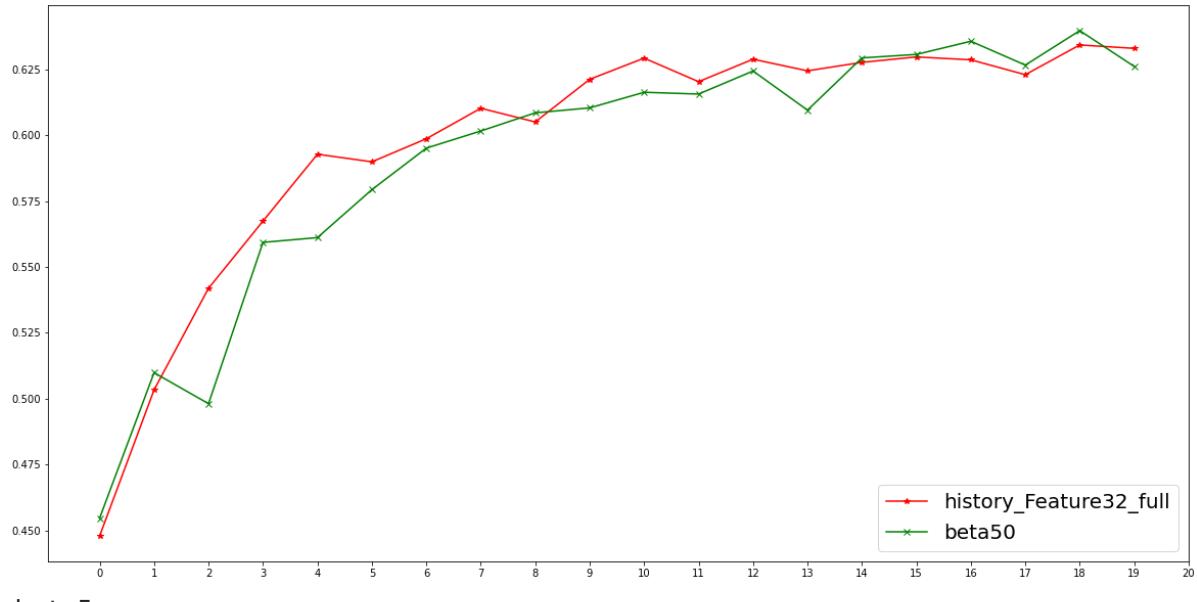
```
In [110]: plt.figure(figsize=(20, 10))

plt.plot(history_Feature32_full["F1_val"], 'r', marker='*', label="history_Feat
plt.plot(beta50["f1_val"], 'g', marker='x', label="beta50")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={"size": 20})

plt.show()

maxf1 = beta50[beta50['f1_val']==max(beta50['f1_val'])]
print("beta5")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(beta50.tail(1))
```



```
beta5
max:
  step      f1_val
18      19  0.639823

last epoch:
  step      f1_val
19      20  0.626229
```

UNET original vs beta70KMeans

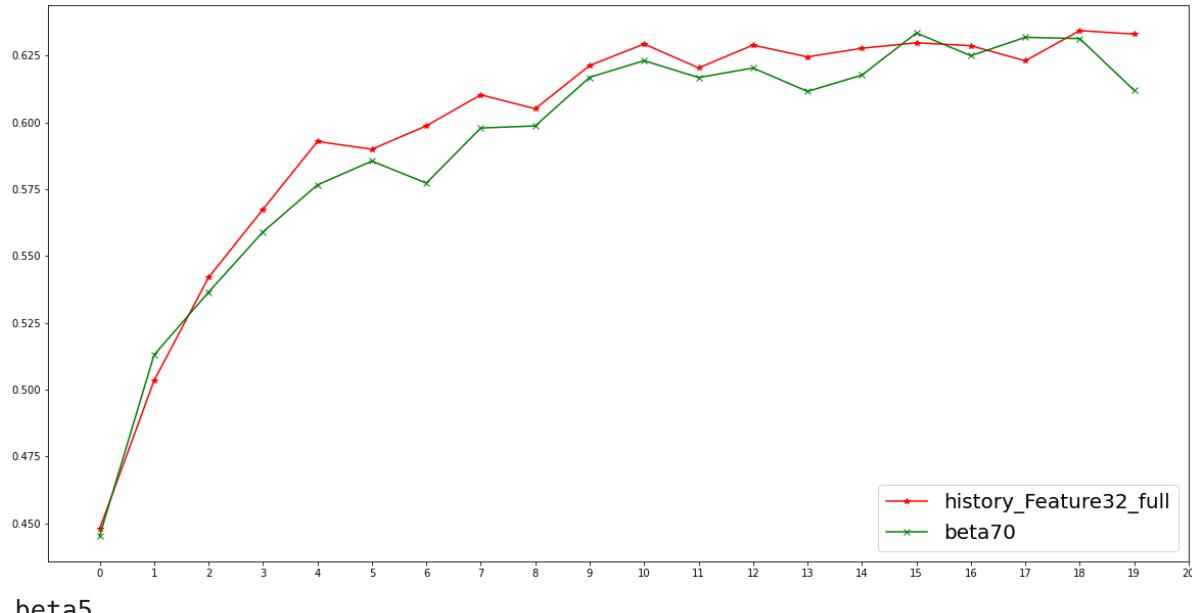
```
In [111]: plt.figure(figsize=(20, 10))

plt.plot(history_Feature32_full["F1_val"], 'r', marker='*', label="history_Feat
plt.plot(beta70["f1_val"], 'g', marker='x', label="beta70")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()

maxf1 = beta70[beta70['f1_val']==max(beta70['f1_val'])]
print("beta5")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(beta70.tail(1))
```



```
beta5
max:
  step      f1_val
15      16  0.633496

last epoch:
  step      f1_val
19      20  0.612022
```

UNET 2 vs RootedRatioLoss

```
In [112]: ### loss = np.sqrt(totalNnc/totalNc)
### this new loss was added to the previous three losses.
### This is the ratio of contour points to non contour points
```

In [113]:

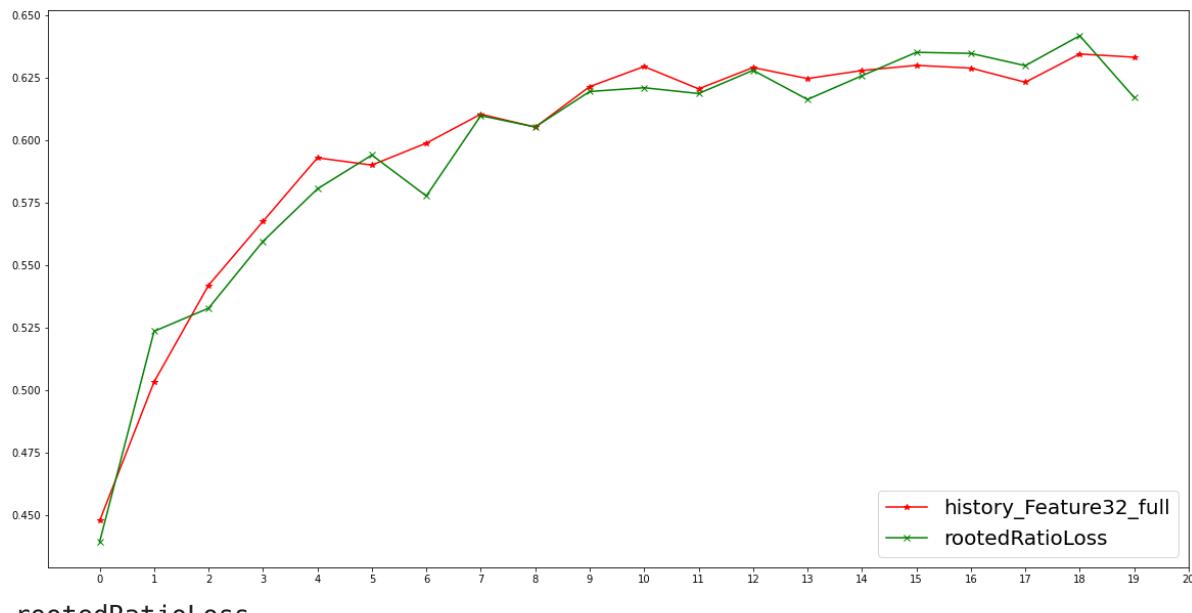
```
plt.figure(figsize=(20, 10))
rootedRatioLoss = pd.read_csv("f1_val_2022-09-06_Exp_RatioLoss.csv")

plt.plot(history_Feature32_full["F1_val"], 'r', marker='*', label="history_Feat")
plt.plot(rootedRatioLoss["f1_val"], 'g', marker='x', label="rootedRatioLoss")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()

maxf1 = rootedRatioLoss[rootedRatioLoss['f1_val']==max(rootedRatioLoss['f1_v
print("rootedRatioLoss")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(rootedRatioLoss.tail(1))
```



rootedRatioLoss

max:

step	f1_val	
18	19	0.641684

last epoch:

step	f1_val	
19	20	0.61699

In [114]:

```
maxf1 = history_Feature32_full[history_Feature32_full['F1_val']==max(history
maxf1
```

Out[114]:

Epoch	F1_val	
18	19	0.634379

UNET 2 vs ReLU inside UP

old UNET: concatenation in the decoder used $\text{RELU}\{\text{encoder layer}_i\} + [\text{transpose convolution decoder layer } i]$

transpose convolution didnt have RELU

Results below are for UNET with $\text{RELU}\{\text{decoder layer } i\}$ before transpose convolution

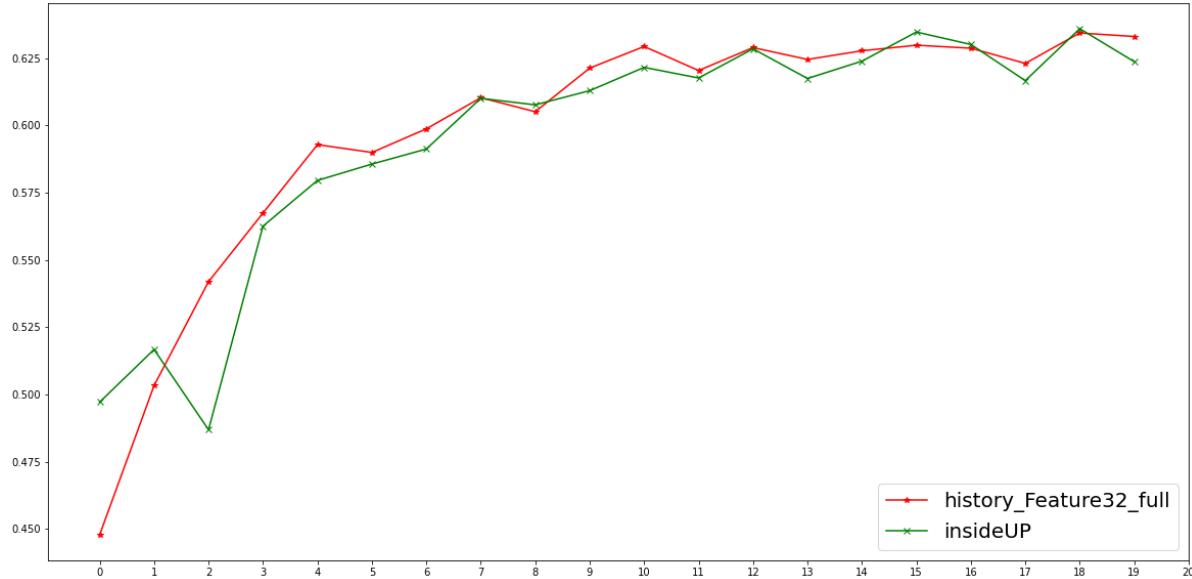
```
In [115...]
plt.figure(figsize=(20, 10))
insideUP = pd.read_csv("f1_val_2022_09_07_ReLU_insideUp.csv")

plt.plot(history_Feature32_full["F1_val"], 'r', marker='*', label="history_Feat")
plt.plot(insideUP["f1_val"], 'g', marker='x', label="insideUP")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()

maxf1 = insideUP[insideUP['f1_val']==max(insideUP['f1_val'])]
print("insideUP")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(insideUP.tail(1))
```



```
insideUP
max:
  step      f1_val
18     19  0.635983

last epoch:
  step      f1_val
19     20  0.623739
```

UNET 2 vs ReLU inside Merge

old UNET: concatenation in the decoder used RELU{encoder layer i} + [transpose convolution decoder layer i]

transpose convolution didnt have RELU

Results below are for UNET with RELU{after transpose convolution}

In [116...]

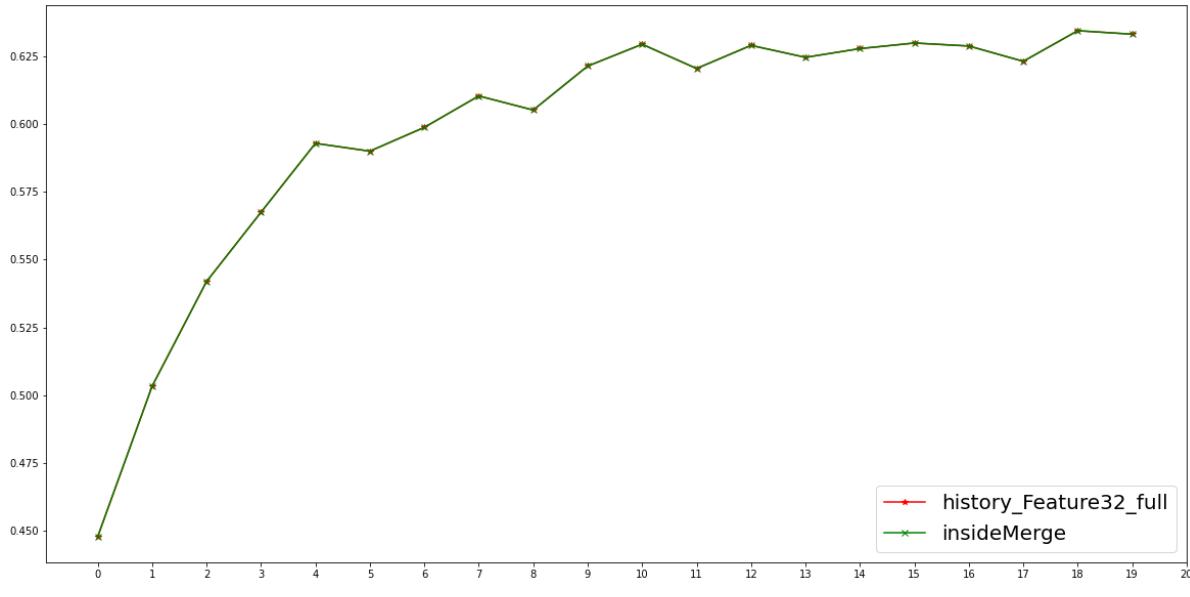
```
plt.figure(figsize=(20, 10))
insideMerge = pd.read_csv("f1_val_2022_09_07_ReluMerge.csv")

plt.plot(history_Feature32_full["F1_val"], 'r', marker='*', label="history_Feat")
plt.plot(insideMerge["f1_val"], 'g', marker='x', label="insideMerge")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()

maxf1 = insideMerge[insideMerge['f1_val']==max(insideMerge['f1_val'])]
print("insideMerge")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(insideMerge.tail(1))
```



```
insideMerge
max:
  step      f1_val
18     19   0.634379

last epoch:
  step      f1_val
19     20   0.633086
```

Experiments on Learning Rate - Warm Up

Experiment 1: First epoch start from alpha/32

and increase by beta every batch. beta=32** (1/285)

later epochs keep lr = alpha = 0.003

In [117...]

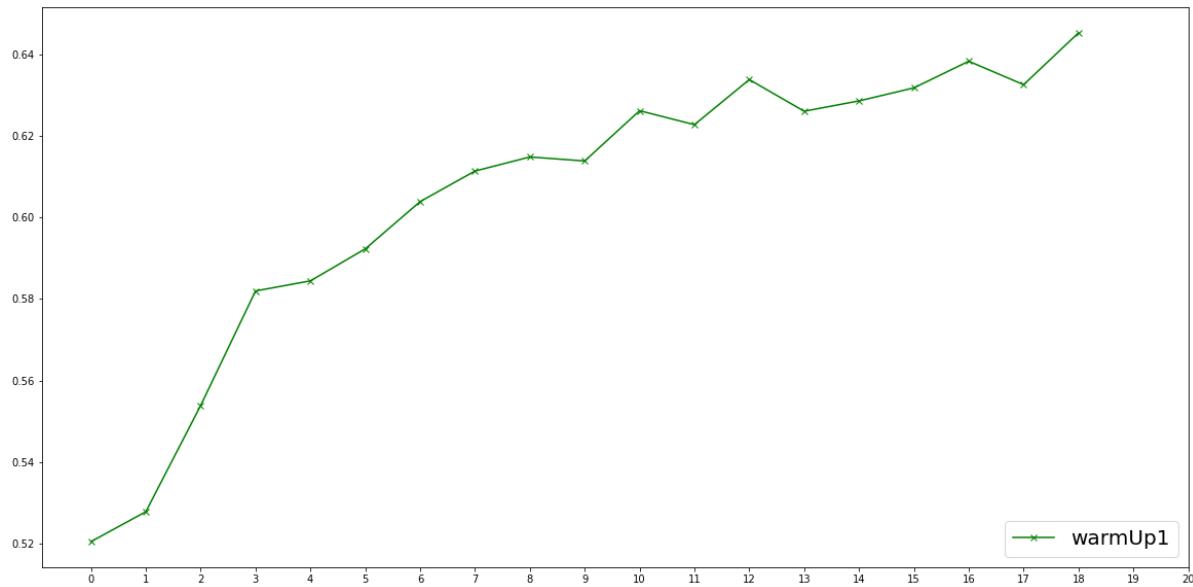
```
plt.figure(figsize=(20, 10))
warmUp1 = pd.read_csv("history_feature32_Warmup_exp1.csv")

# plt.plot(history_Feature32_full["F1_val"],'r',marker='*',label="history_Fe
plt.plot(warmUp1["f1_val"],'g',marker='x',label="warmUp1")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()

maxf1 = warmUp1[warmUp1['f1_val']==max(warmUp1['f1_val'])]
print("warmUp1")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(warmUp1.tail(1))
```



warmUp1

max:

step	f1_val	
18	19	0.645271

last epoch:

step	f1_val	
18	19	0.645271

Experiment 2: First epoch start from alpha/32

From epoch 2 to end decrease from alpha by rate of beta

beta = 100**(-1/20)

In [118...]

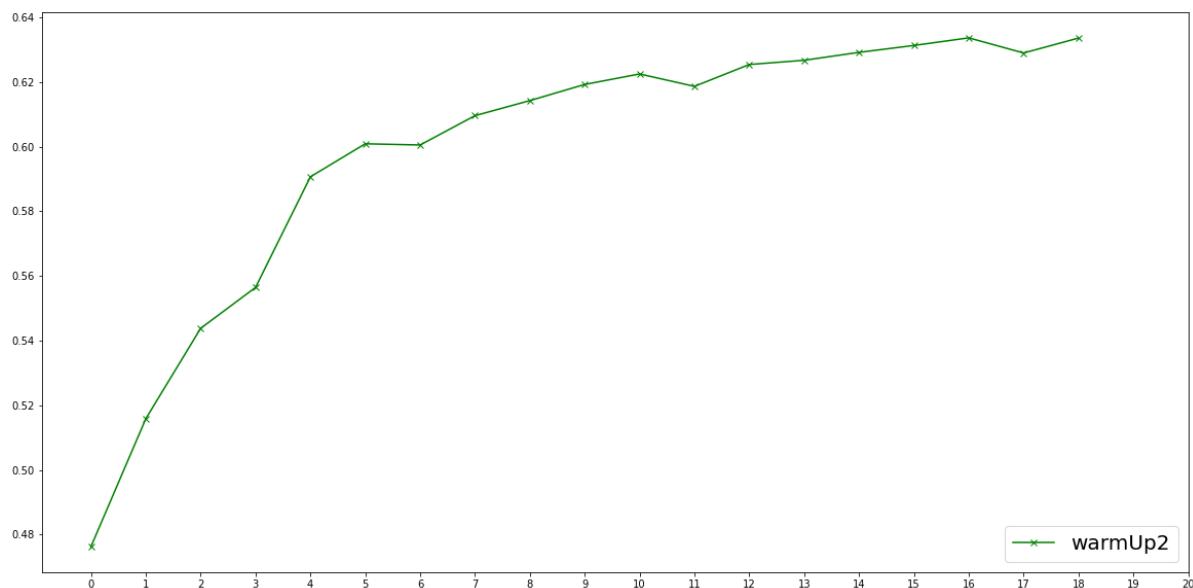
```
plt.figure(figsize=(20, 10))
warmUp2 = pd.read_csv("history_feature32_Warmup_exp2.csv")

# plt.plot(history_Feature32_full["F1_val"],'r',marker='*',label="history_Fe
plt.plot(warmUp2["f1_val"],'g',marker='x',label="warmUp2")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()

maxf1 = warmUp2[warmUp2['f1_val']==max(warmUp2['f1_val'])]
print("warmUp2")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(warmUp2.tail(1))
```



```
warmUp2
max:
  step      f1_val
16      17    0.633709

last epoch:
  step      f1_val
18      19    0.633646
```

Experiment 3: First epoch start from alpha/32

and increase by beta every batch.

beta=32** $(1/285)$

Drop by 10 at epoch 10 and again at epoch 15

In [119]:

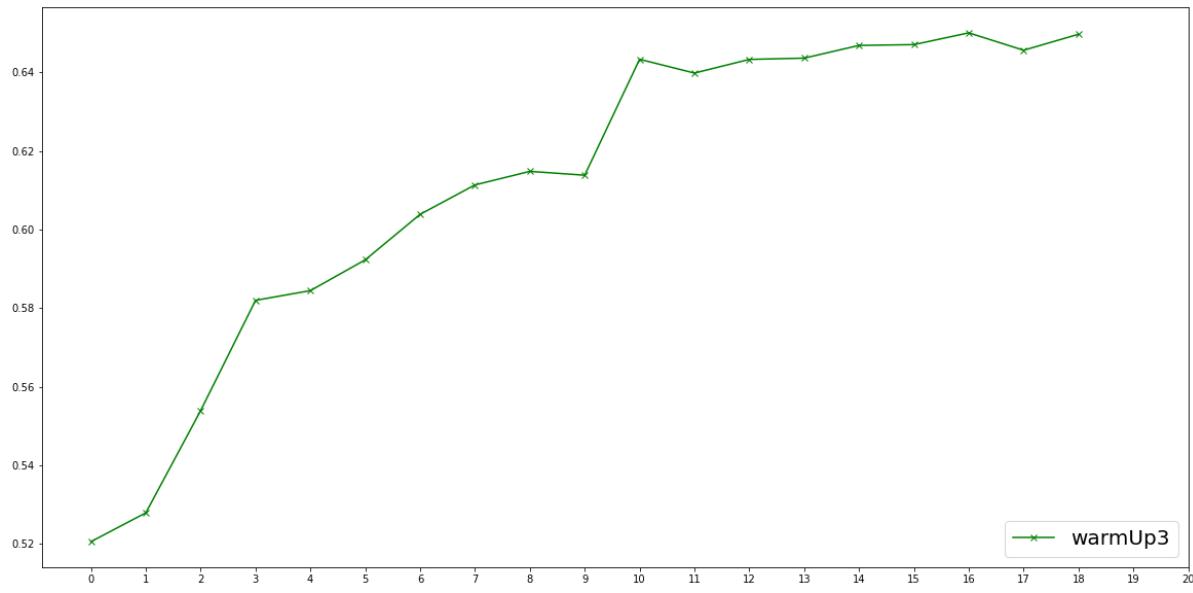
```
plt.figure(figsize=(20, 10))
warmUp3 = pd.read_csv("history_feature32_Warmup_exp3.csv")

# plt.plot(history_Feature32_full["F1_val"], 'r', marker='*', label="history_Fe
plt.plot(warmUp3["f1_val"], 'g', marker='x', label="warmUp3")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()

maxf1 = warmUp3[warmUp3['f1_val']==max(warmUp3['f1_val'])]
print("warmUp3")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(warmUp3.tail(1))
```



warmUp3

max:

step	f1_val	
16	17	0.65012

last epoch:

step	f1_val	
18	19	0.649758

Experiment 3: First epoch start from alpha/32

and increase by beta every batch.

beta=32** $(1/285)$

From epoch 2 onwards reduce lr by beta = $100^{**}(-1/20)$

In [120]:

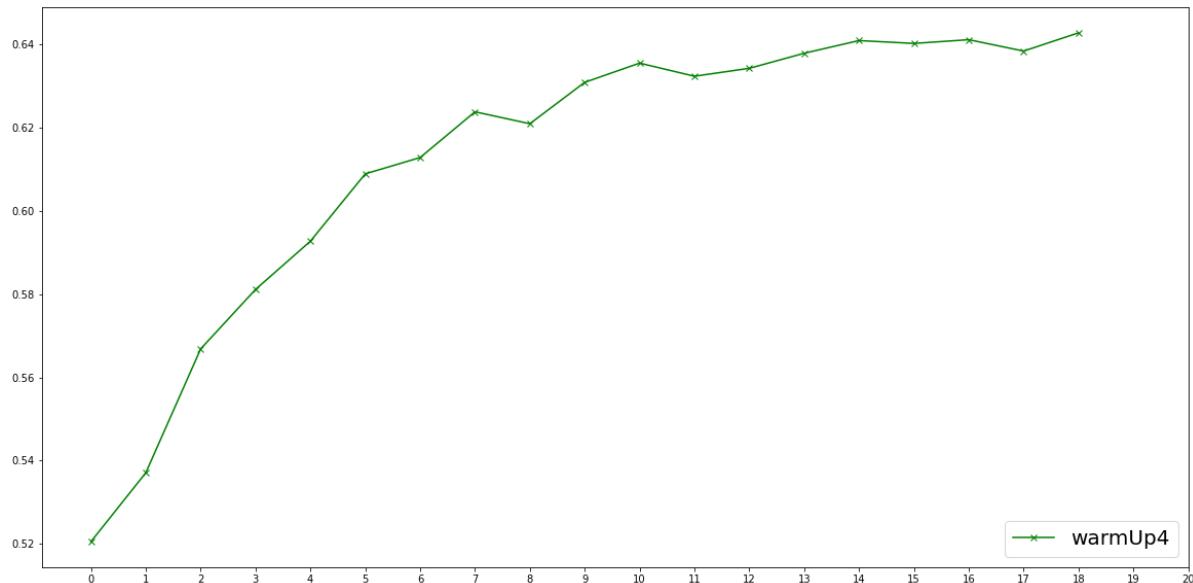
```
plt.figure(figsize=(20, 10))
warmUp4 = pd.read_csv("history_feature32_Warmup_exp4.csv")

# plt.plot(history_Feature32_full["F1_val"], 'r', marker='*', label="history_Feature32_full")
plt.plot(warmUp4["f1_val"], 'g', marker='x', label="warmUp4")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()

maxf1 = warmUp4[warmUp4['f1_val']==max(warmUp4['f1_val'])]
print("warmUp4")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(warmUp4.tail(1))
```



warmUp4

max:

step	f1_val	
18	19	0.642806

last epoch:

step	f1_val	
18	19	0.642806

ATTENTION EXPERIMENTS

All experiments

Experiment 1: Attention with gating signal

the multiplication upsampled using transpose convolution and filter size 2

```
In [121]: plt.figure(figsize=(20, 10))
conv2dAttention = pd.read_csv("UNET_full_9k_Feature32_Attention-ConvTranspose.csv")

plt.plot(conv2dAttention["f1_val"], 'g', marker='x', label="conv2dAttention")
plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})
plt.show()

maxf1 = conv2dAttention[conv2dAttention['f1_val']==max(conv2dAttention['f1_val'])]
print("conv2dAttention")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(conv2dAttention.tail(1))
```



```
conv2dAttention
max:
    step      f1_val
17      18  0.505993

last epoch:
    step      f1_val
19      20  0.499013
```

Experiment 2: Attention with gating signal

the multiplication upsampled using upsample method mode nearest

```
nn.Upsample(scale_factor=2, mode='nearest').to(device)
```

In [122]:

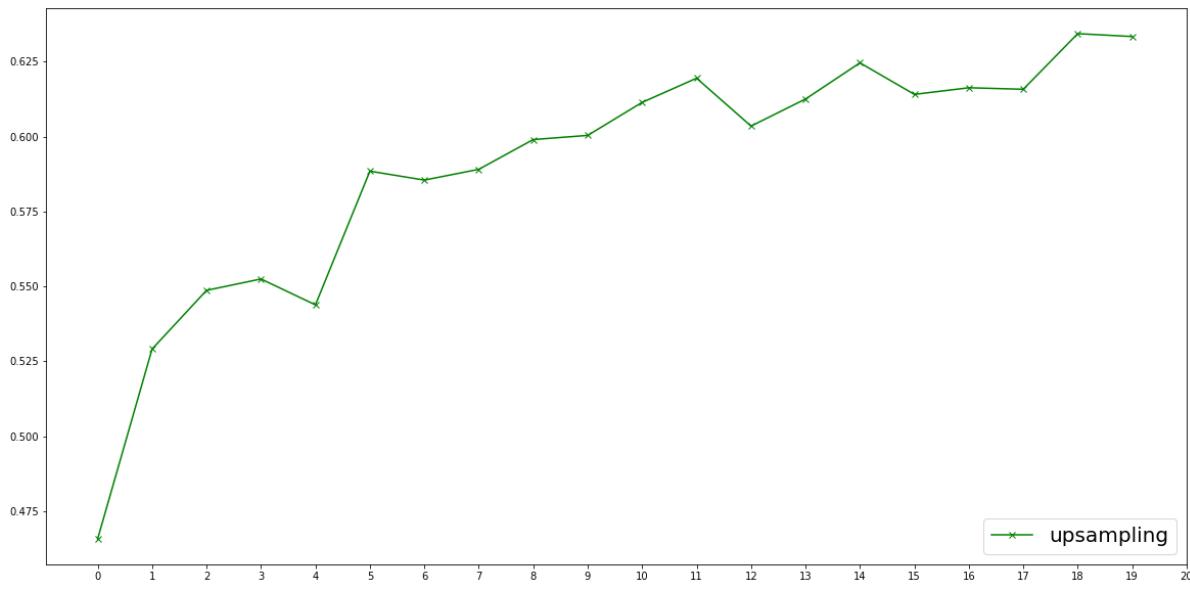
```
plt.figure(figsize=(20, 10))
upsampling = pd.read_csv("UNET_full_9k_Feature32_Attention_upsampling_f1_val.csv")

plt.plot(upsampling["f1_val"], 'g', marker='x', label="upsampling")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()

maxf1 = upsampling[upsampling['f1_val']==max(upsampling['f1_val'])]
print("upsampling")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(upsampling.tail(1))
```



upsampling

max:

```
    step      f1_val
18      19    0.634398
```

last epoch:

```
    step      f1_val
19      20    0.633428
```

Experiment 3: Attention with gating signal

the multiplication upsampled using upsample method Bilinear

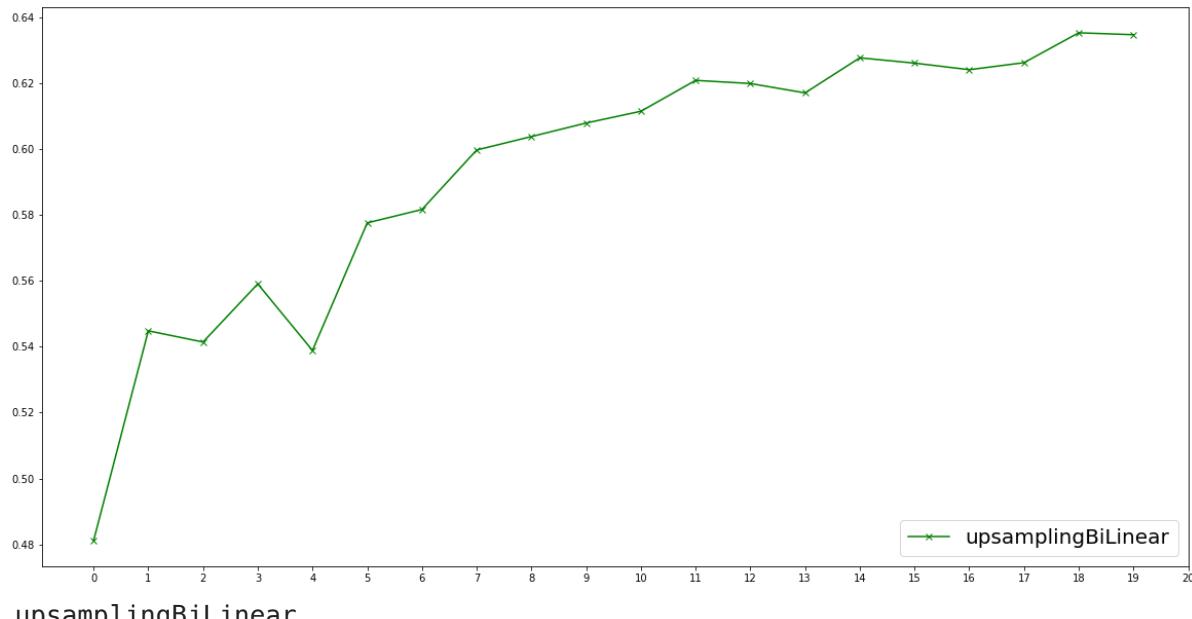
In [123...]

```
plt.figure(figsize=(20, 10))
upsamplingBiLinear = pd.read_csv("UNET_full_9k_Feature32_Attention-UpsampleE

plt.plot(upsamplingBiLinear["f1_val"], 'g', marker='x', label="upsamplingBiLinear")
plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()

maxf1 = upsamplingBiLinear[upsamplingBiLinear['f1_val']==max(upsamplingBiLinear['f1_val'])]
print("upsamplingBiLinear")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(upsamplingBiLinear.tail(1))
```



upsamplingBiLinear
max:

```
      step    f1_val
18      19  0.635215
```

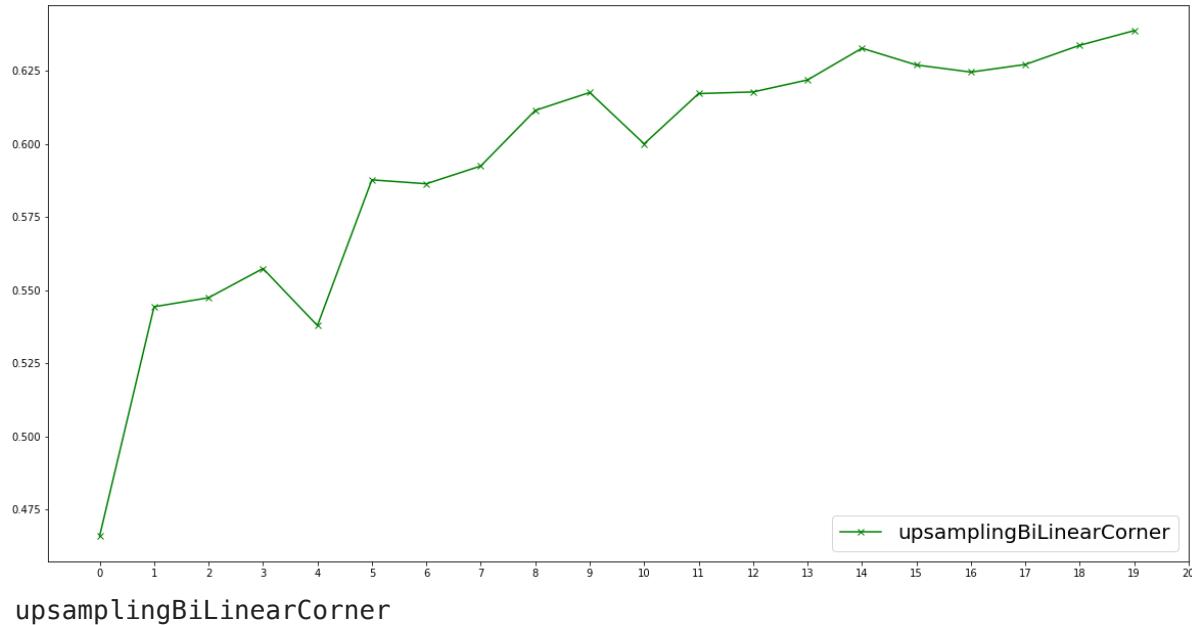
last epoch:
 step f1_val
19 20 0.634636

Experiment 4: Attention with gating signal

the multiplication upsampled using upsample method
Bilinear_align_corners

```
In [124... plt.figure(figsize=(20, 10))
upsamplingBiLinearCorner= pd.read_csv("UNET_full_9k_Feature32_Attention-Upsa
plt.plot(upsamplingBiLinearCorner["f1_val"],'g',marker='x',label="upsamplingBiL
plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})
plt.show()

maxf1 = upsamplingBiLinearCorner[upsamplingBiLinearCorner['f1_val']==max(upsam
print("upsamplingBiLinearCorner")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(upsamplingBiLinearCorner.tail(1))
```



```
upsamplingBiLinearCorner
max:
```

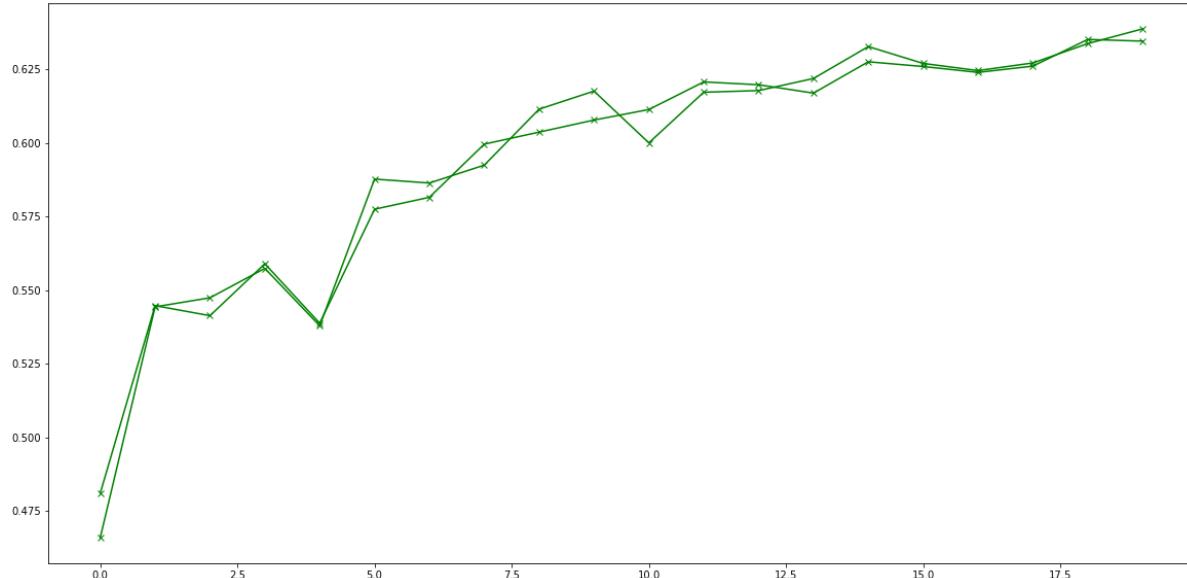
```
    step      f1_val
19      20  0.638807
```

```
last epoch:
    step      f1_val
19      20  0.638807
```

```
In [125... plt.figure(figsize=(20, 10))
```

```
plt.plot(upsamplingBiLinearCorner["f1_val"],'g',marker='x',label="upsamplingBiL
plt.plot(upsamplingBiLinear["f1_val"],'g',marker='x',label="upsamplingBiLine
```

```
Out[125]: [<matplotlib.lines.Line2D at 0x1fc78d2df10>]
```



CBAM Experiment

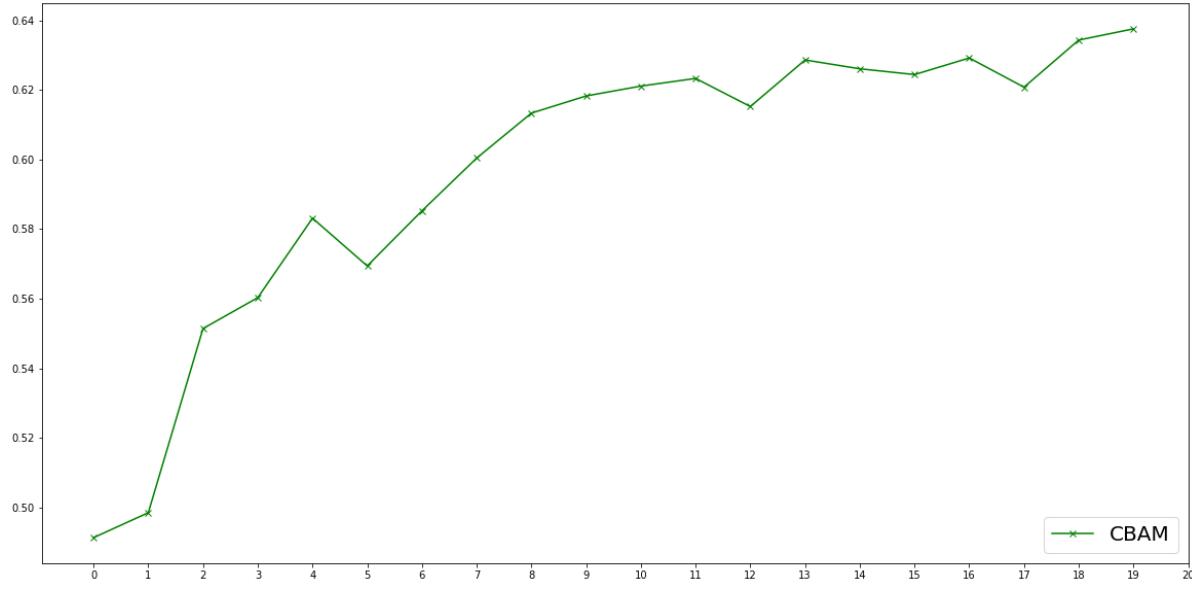
```
In [126]: plt.figure(figsize=(20, 10))
CBAM= pd.read_csv("UNET_full_9k_Feature32_Attention-CBAM_f1_val.csv")

plt.plot(CBAM["f1_val"],'g',marker='x',label="CBAM")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()

maxf1 = CBAM[CBAM['f1_val']==max(CBAM['f1_val'])]
print("CBAM")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(CBAM.tail(1))
```



CBAM

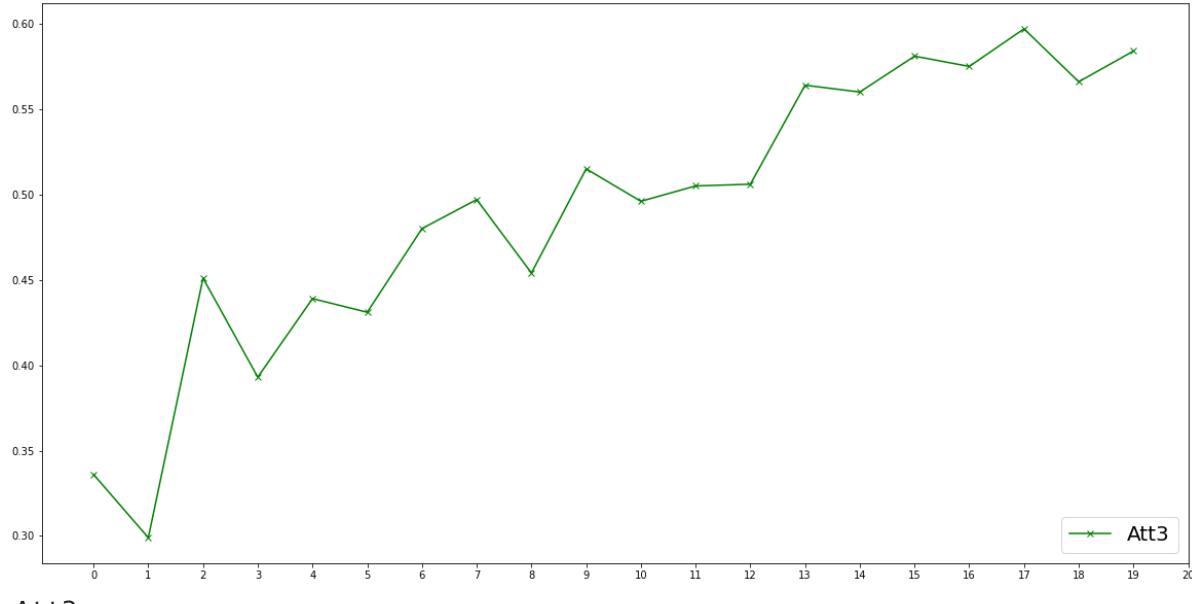
```
max:  
    step      f1_val  
19      20   0.637523  
  
last epoch:  
    step      f1_val  
19      20   0.637523
```

Attention Experiments

```
In [127]: ## attention applied to combination of encoder output and output from decoder  
## convolution transpose using kernel size 3
```



```
In [128]: # UNET_full_9k_Feature32_Attention-ConvTransposeBy3.csv  
  
plt.figure(figsize=(20, 10))  
Att3= pd.read_csv("UNET_full_9k_Feature32_Attention-ConvTransposeBy3.csv")  
  
plt.plot(Att3["f1_val"],'g',marker='x',label="Att3")  
  
plt.xticks(range(0,21))  
plt.legend(loc="lower right", prop={'size': 20})  
  
plt.show()  
  
maxf1 = Att3[Att3['f1_val']==max(Att3['f1_val'])]  
print("Att3")  
print("max:")  
print(maxf1)  
print()  
print("last epoch:")  
print(Att3.tail(1))
```



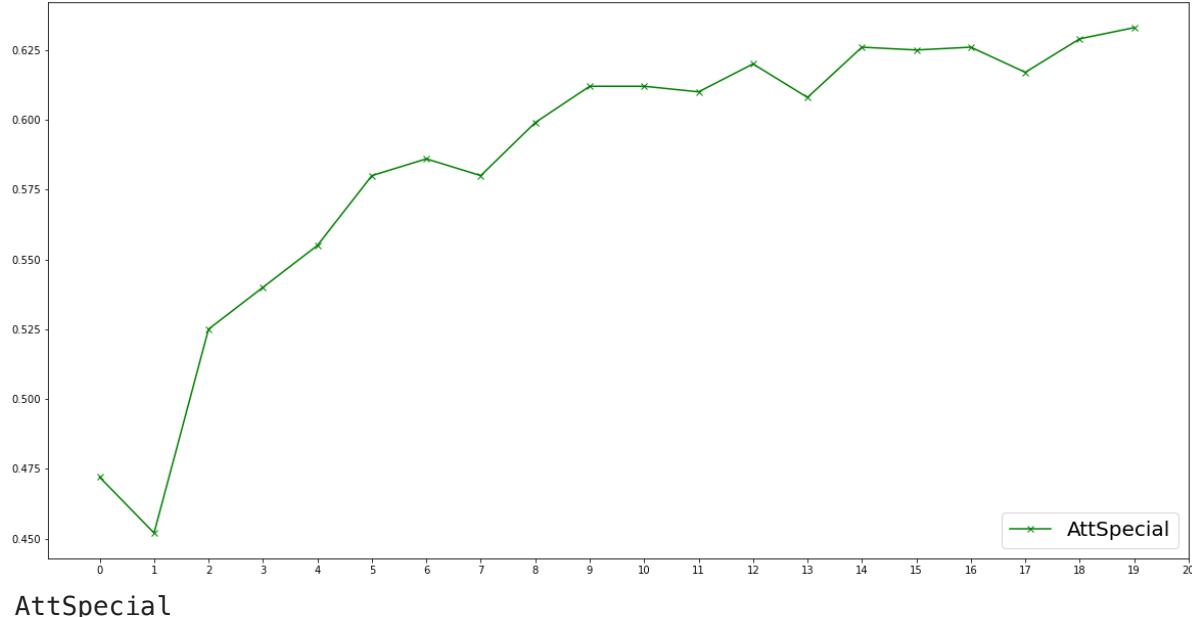
Att3

```
max:  
    step  f1_val  
17      17    0.597
```

```
last epoch:  
    step  f1_val  
19      19    0.584
```

```
In [129]: # Spatial and Channel Attention (Channel Attention modified according to Pr  
# Pooling
```

```
In [130]: plt.figure(figsize=(20, 10))  
AttSpecial= pd.read_csv("UNET_full_9k_Feature32_Attention_specialized.csv")  
  
plt.plot(AttSpecial["f1_val"], 'g', marker='x', label="AttSpecial")  
  
plt.xticks(range(0,21))  
plt.legend(loc="lower right", prop={'size': 20})  
  
plt.show()  
  
maxf1 = AttSpecial[AttSpecial['f1_val']==max(AttSpecial['f1_val'])]  
print("AttSpecial")  
print("max:")  
print(maxf1)  
print()  
print("last epoch:")  
print(AttSpecial.tail(1))
```



AttSpecial

max:

 Unnamed: 0 f1_val
19 19 0.633

last epoch:

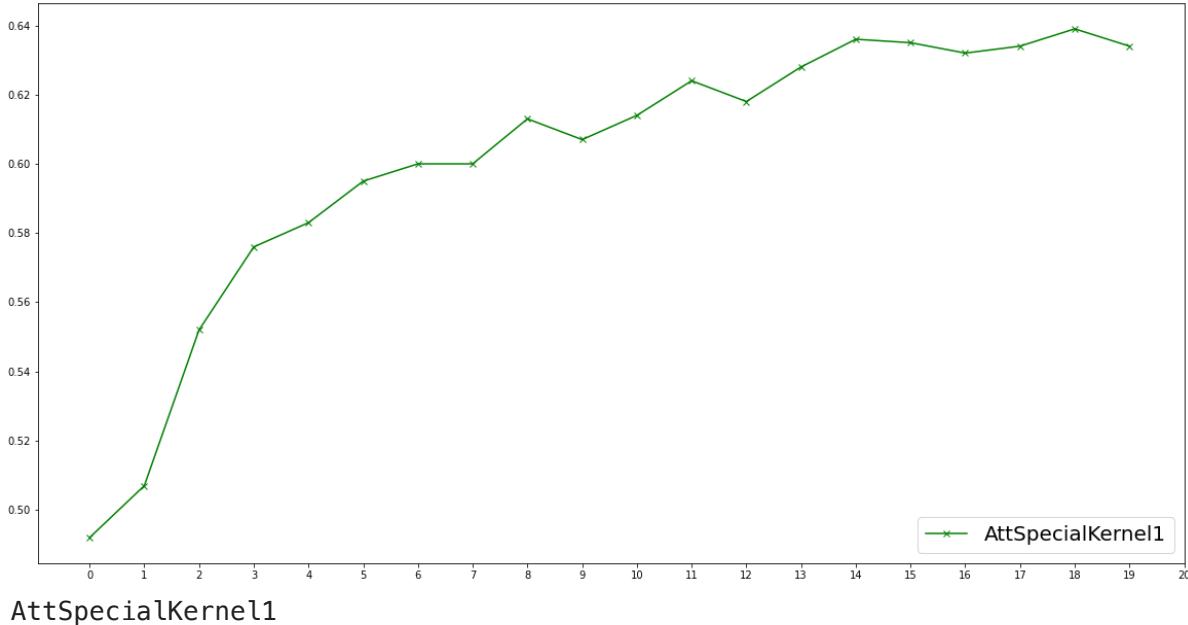
 Unnamed: 0 f1_val
19 19 0.633

In [131]: # Continuing with Special Attention Filter. Now kernel size for spatial filter

```
In [132]: # UNET_full_9k_Feature32_Attention_specialized_filter0f1
plt.figure(figsize=(20, 10))
AttSpecialKernel1= pd.read_csv("UNET_full_9k_Feature32_Attention_specialized_
filter0f1.csv")

plt.plot(AttSpecialKernel1["f1_val"], 'g', marker='x', label="AttSpecialKernel1")
plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})
plt.show()

maxf1 = AttSpecialKernel1[AttSpecialKernel1['f1_val']==max(AttSpecialKernel1['f1_val'])]
print("AttSpecialKernel1")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(AttSpecialKernel1.tail(1))
```



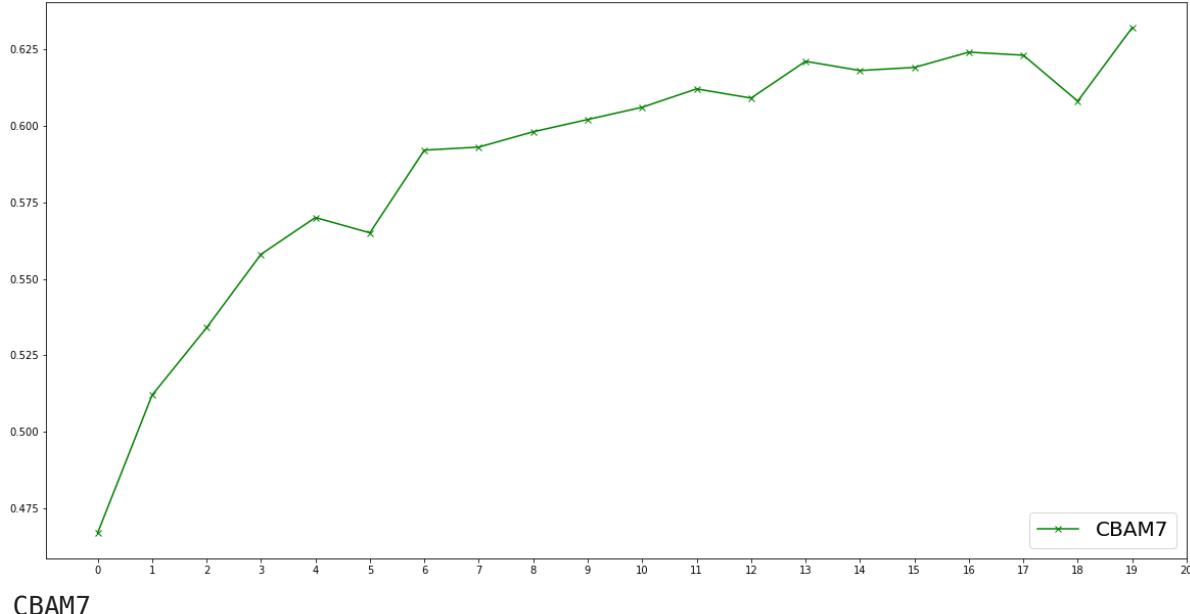
```
AttSpecialKernel1  
max:  
    Unnamed: 0   f1_val  
18          18   0.639
```

```
last epoch:  
    Unnamed: 0   f1_val  
19          19   0.634
```

```
In [133...]: # CBAM with kernel size 7
```

```
In [134...]: # UNET_full_9k_Feature32_Attention-CBAM-FilterSize7
```

```
plt.figure(figsize=(20, 10))  
CBAM7= pd.read_csv("UNET_full_9k_Feature32_Attention-CBAM-FilterSize7.csv")  
  
plt.plot(CBAM7["f1_val"], 'g', marker='x', label="CBAM7")  
  
plt.xticks(range(0,21))  
plt.legend(loc="lower right", prop={'size': 20})  
  
plt.show()  
  
maxf1 = CBAM7[CBAM7['f1_val']==max(CBAM7['f1_val'])]  
print("CBAM7")  
print("max:")  
print(maxf1)  
print()  
print("last epoch:")  
print(CBAM7.tail(1))
```



CBAM7

max:

Unnamed: 0	f1_val	
19	19	0.632

last epoch:

Unnamed: 0	f1_val	
19	19	0.632

In [135...]: # CBAM with filter size 1

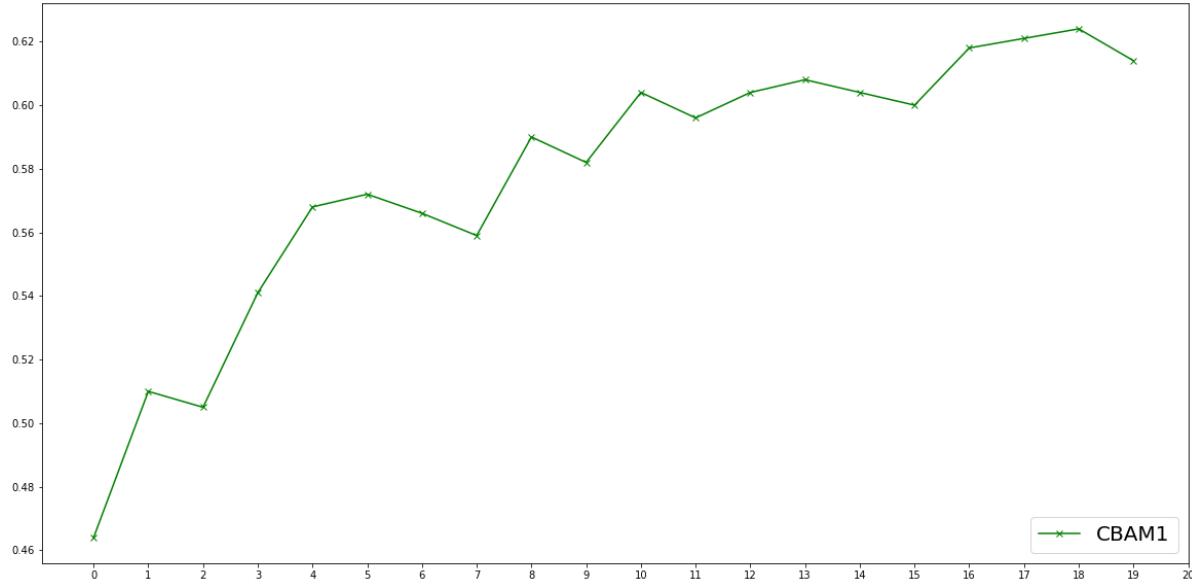
```
# UNET_full_9k_Feature32_Attention-CBAM-FilterSize1
plt.figure(figsize=(20, 10))
CBAM1= pd.read_csv("UNET_full_9k_Feature32_Attention-CBAM_filterSize1.csv")

plt.plot(CBAM1["f1_val"], 'g', marker='x', label="CBAM1")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()

maxf1 = CBAM1[CBAM1['f1_val']==max(CBAM1['f1_val'])]
print("CBAM1")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(CBAM1.tail(1))
```



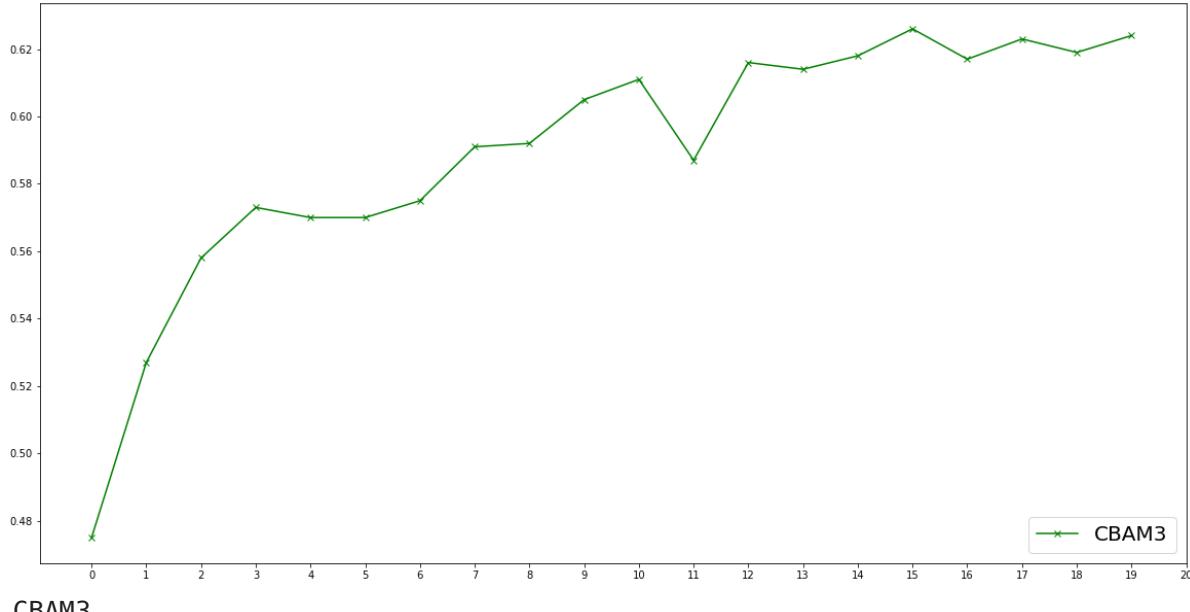
CBAM1

```
max:  
    Unnamed: 0    f1_val  
18          18    0.624
```

```
last epoch:  
    Unnamed: 0    f1_val  
19          19    0.614
```

```
In [137...]: # CBAM with filter size 3
```

```
In [138...]: # UNET_full_9k_Feature32_Attention-CBAM-FilterSize1  
plt.figure(figsize=(20, 10))  
CBAM3= pd.read_csv("UNET_full_9k_Feature32_Attention-CBAM-FilterSize3.csv")  
  
plt.plot(CBAM3["f1_val"], 'g', marker='x', label="CBAM3")  
  
plt.xticks(range(0,21))  
plt.legend(loc="lower right", prop={'size': 20})  
plt.show()  
  
maxf1 = CBAM3[CBAM3['f1_val']==max(CBAM3['f1_val'])]  
print("CBAM3")  
print("max:")  
print(maxf1)  
print()  
print("last epoch:")  
print(CBAM3.tail(1))
```

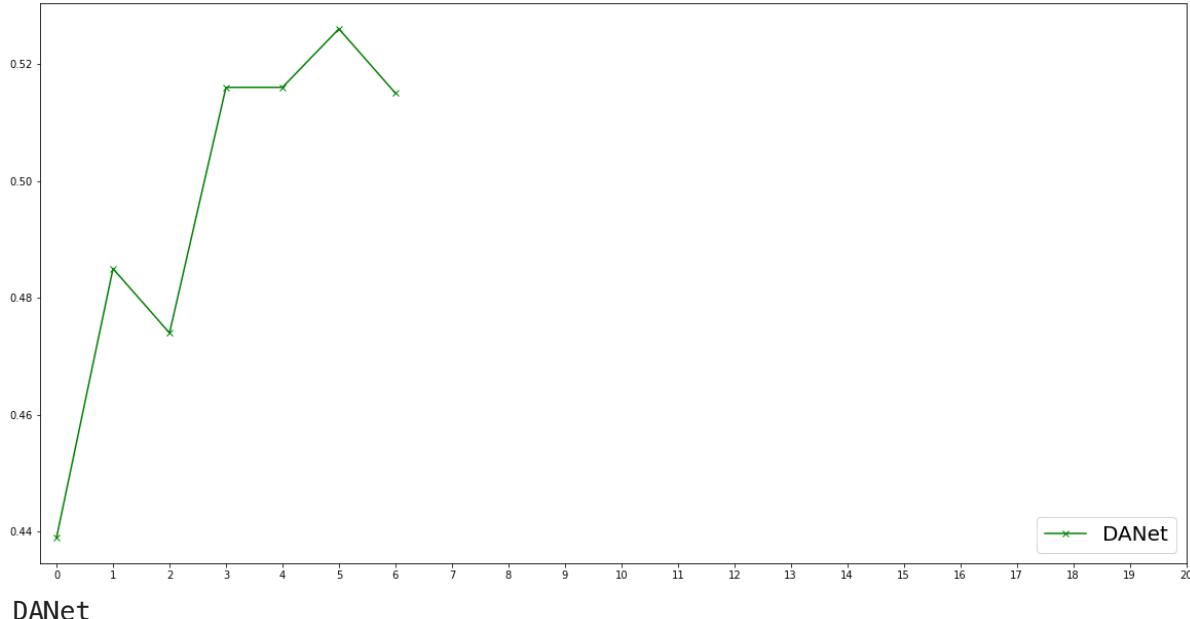


CBAM3

```
max:  
    Unnamed: 0   f1_val  
15          15   0.626  
  
last epoch:  
    Unnamed: 0   f1_val  
19          19   0.624
```

```
In [139]: #DANet Incomplete Result  
# UNET_full_9k_Feature32_DANet-ResultBackUp  
# Use of DANet on skip connections
```

```
In [140]: plt.figure(figsize=(20, 10))  
DANet= pd.read_csv("UNET_full_9k_Feature32_DANet-ResultBackUp.csv")  
  
plt.plot(DANet["f1_val"], 'g', marker='x', label="DANet")  
  
plt.xticks(range(0,21))  
plt.legend(loc="lower right", prop={'size': 20})  
  
plt.show()  
  
maxf1 = DANet[DANet['f1_val']==max(DANet['f1_val'])]  
print("DANet")  
print("max:")  
print(maxf1)  
print()  
print("last epoch:")  
print(DANet.tail(1))
```



```
DANet
max:
    Unnamed: 0   f1_val
5           5   0.526

last epoch:
    Unnamed: 0   f1_val
6           6   0.515
```

Y different Scale Experiments

Y was downscaled to various sizes in the decoder layer and used to calculate layer losses

```
In [141]: # UNET_full_9k_Feature32_AttentionLoss-Copy1
# Use of Attention Loss of each decoder layer with downsampled label y
```

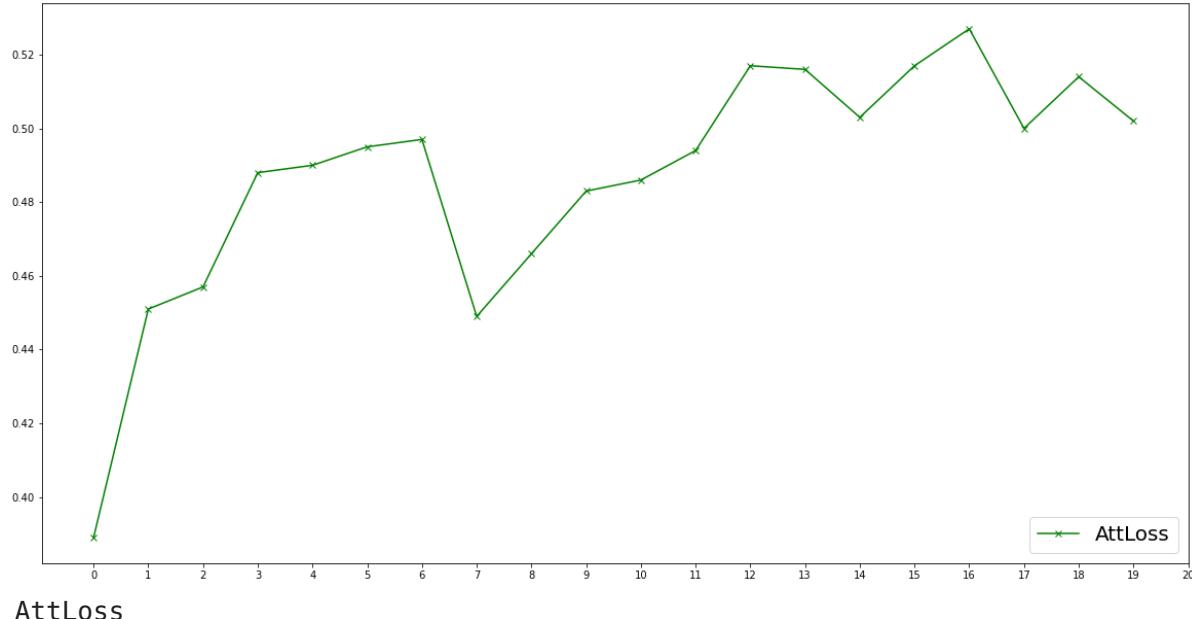
```
In [142]: plt.figure(figsize=(20, 10))
AttLoss= pd.read_csv("UNET_full_9k_Feature32_AttentionLoss-Copy1.csv")

plt.plot(AttLoss["f1_val"],'g',marker='x',label="AttLoss")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()

maxf1 = AttLoss[AttLoss['f1_val']==max(AttLoss['f1_val'])]
print("AttLoss")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(AttLoss.tail(1))
```



```
AttLoss
max:
    Unnamed: 0   f1_val
16           16   0.527

last epoch:
    Unnamed: 0   f1_val
19           19   0.502
```

```
In [143]: # multilayer Channel Concatenation
# combinedOutFull = torch.cat([original,up1_scaled,up2_scaled,up3_scaled], dim=1)

# all channels were added before convolving to single channel
```

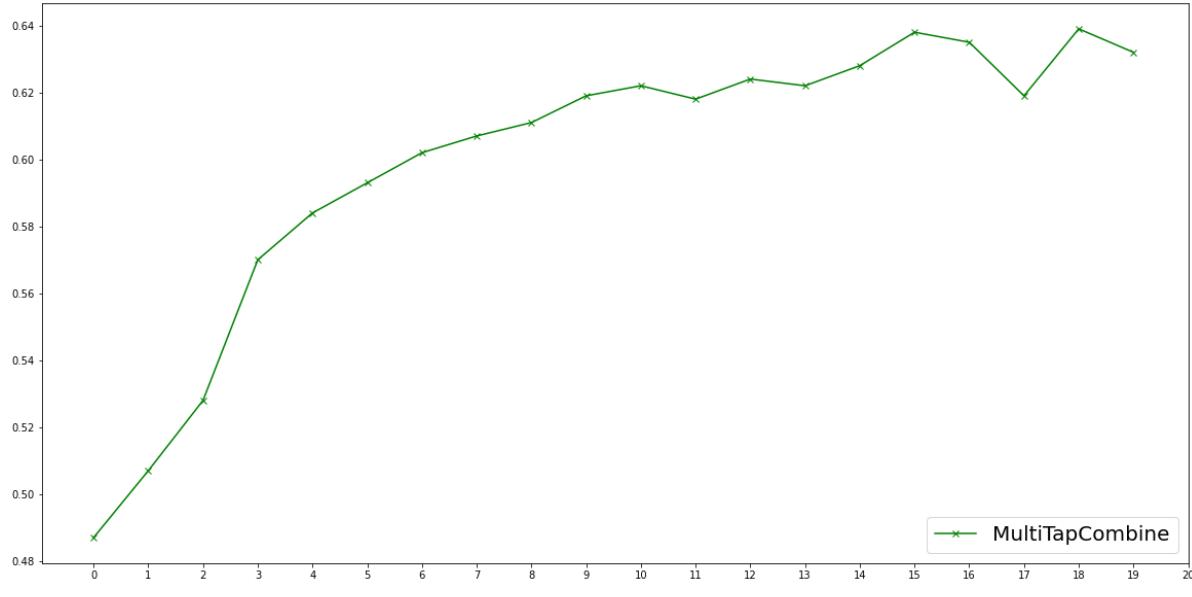
```
In [144]: plt.figure(figsize=(20, 10))
MultiTapCombine= pd.read_csv("UNET_full_9k_Feature32_MultiTap_ChannelConcat.csv")

plt.plot(MultiTapCombine["f1_val"],'g',marker='x',label="MultiTapCombine")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()

maxf1 = MultiTapCombine[MultiTapCombine['f1_val']==max(MultiTapCombine['f1_val'])]
print("MultiTapCombine")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(MultiTapCombine.tail(1))
```

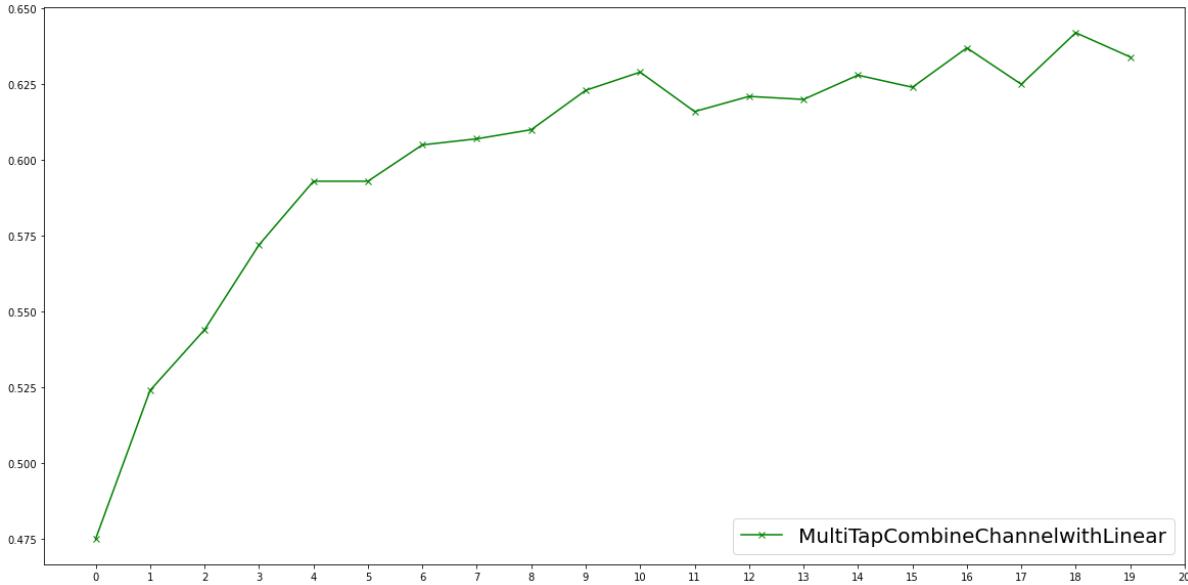


```
MultiTapCombine
```

```
max:  
    Unnamed: 0    f1_val  
18          18    0.639  
  
last epoch:  
    Unnamed: 0    f1_val  
19          19    0.632
```

```
In [145...]: # UNET_full_9k_Feature32_MultiTap_ChannelConcat_Linear  
# all channels were added before using linear to single channel (instead of
```

```
In [146...]: plt.figure(figsize=(20, 10))  
MultiTapCombineChannelwithLinear= pd.read_csv("UNET_full_9k_Feature32_MultiT...  
  
plt.plot(MultiTapCombineChannelwithLinear["f1_val"], 'g', marker='x', label="Mu...  
plt.xticks(range(0,21))  
plt.legend(loc="lower right", prop={'size': 20})  
plt.show()  
  
maxf1 = MultiTapCombineChannelwithLinear[MultiTapCombineChannelwithLinear['f1_v...  
print("MultiTapCombineChannelwithLinear")  
print("max:")  
print(maxf1)  
print()  
print("last epoch:")  
print(MultiTapCombineChannelwithLinear.tail(1))
```



```
MultiTapCombineChannelwithLinear
```

```
max:
```

```
    Unnamed: 0    f1_val  
18          18    0.642
```

```
last epoch:
```

```
    Unnamed: 0    f1_val  
19          19    0.634
```

```
In [147...]: # UNET_full_9k_Feature32_MultiTap
```

```
# Convert decoder outputs to 1 channel before combining to real output  
# final output is linearly added to give 1 channel
```

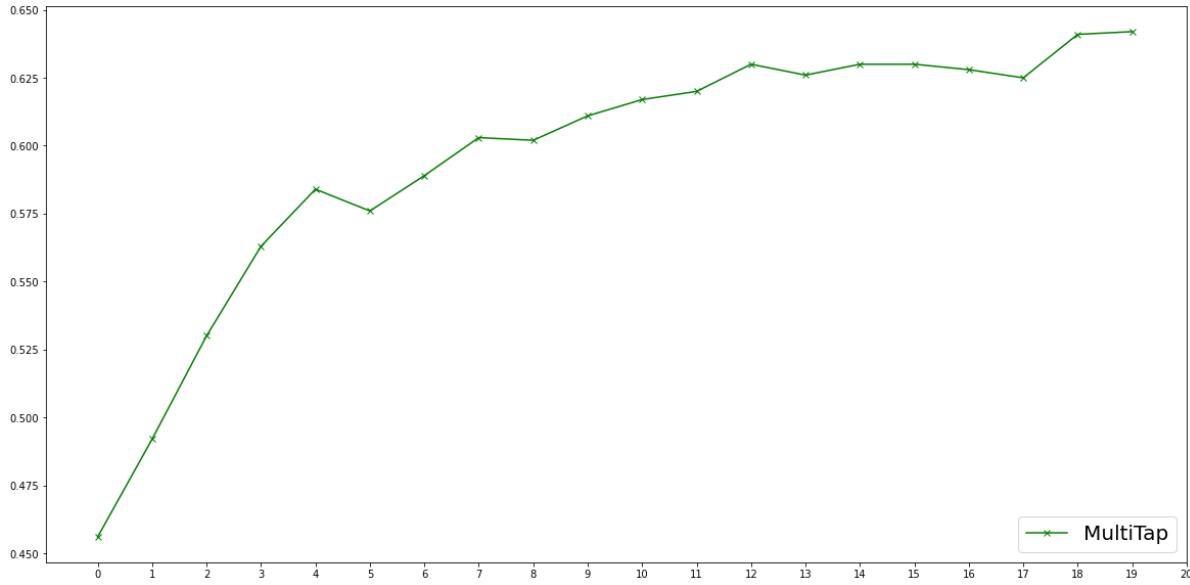
```
In [148...]: plt.figure(figsize=(20, 10))  
MultiTap= pd.read_csv("UNET_full_9k_Feature32_MultiTap.csv")
```

```
plt.plot(MultiTap["f1_val"],'g',marker='x',label="MultiTap")
```

```
plt.xticks(range(0,21))  
plt.legend(loc="lower right", prop={'size': 20})
```

```
plt.show()
```

```
maxf1 = MultiTap[MultiTap['f1_val']==max(MultiTap['f1_val'])]  
print("MultiTap")  
print("max:")  
print(maxf1)  
print()  
print("last epoch:")  
print(MultiTap.tail(1))
```



MultiTap

max:

Unnamed: 0	f1_val	
19	19	0.642

last epoch:

Unnamed: 0	f1_val	
19	19	0.642

In [149]:

```
# UNET_full_9k_Feature32_MultiTap_Conv

# Convert decoder outputs to 1 channel before combining to real output

# final output is convolved to give 1 channel
```

In [150]:

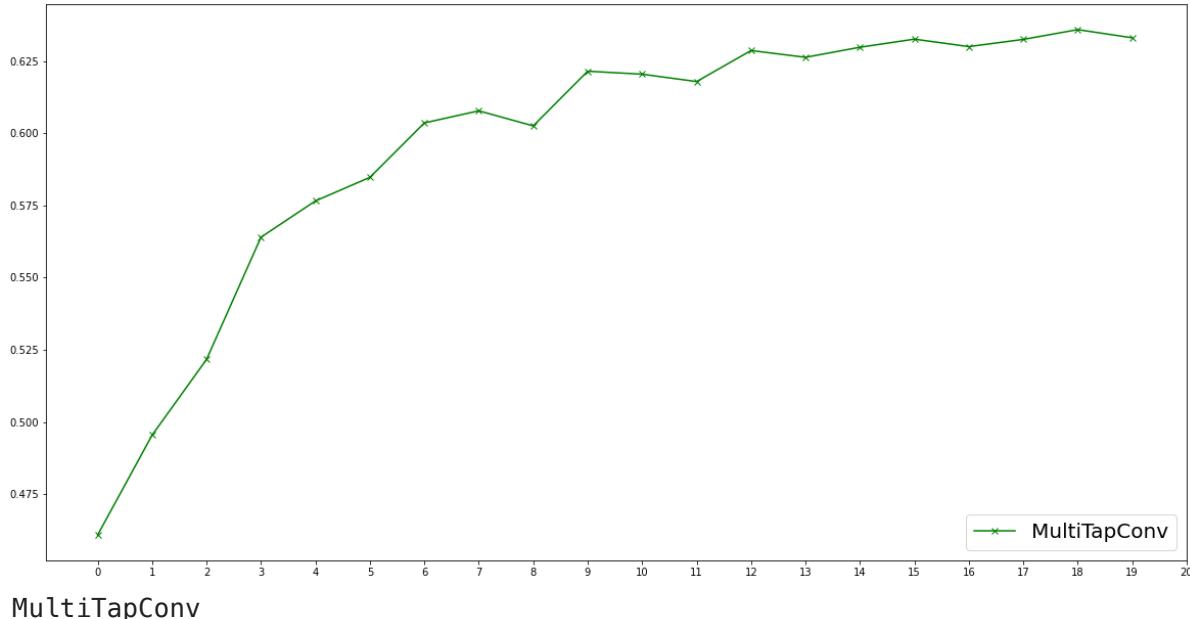
```
plt.figure(figsize=(20, 10))
MultiTapConv = pd.read_csv("UNET_full_9k_Feature32_MultiTap_Conv.csv")

plt.plot(MultiTapConv["f1_val"], 'g', marker='x', label="MultiTapConv")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()

maxf1 = MultiTapConv[MultiTapConv['f1_val'] == max(MultiTapConv['f1_val'])]
print("MultiTapConv")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(MultiTapConv.tail(1))
```



MultiTapConv

max:

Unnamed: 0	f1_val	
18	19	0.635841

last epoch:

Unnamed: 0	f1_val	
19	20	0.63302

In [151]...

```
# UNET_full_9k_Feature32_MultiTap_4Losses
# use each decoder upscaled output to calculate the losses
```

In [152]...

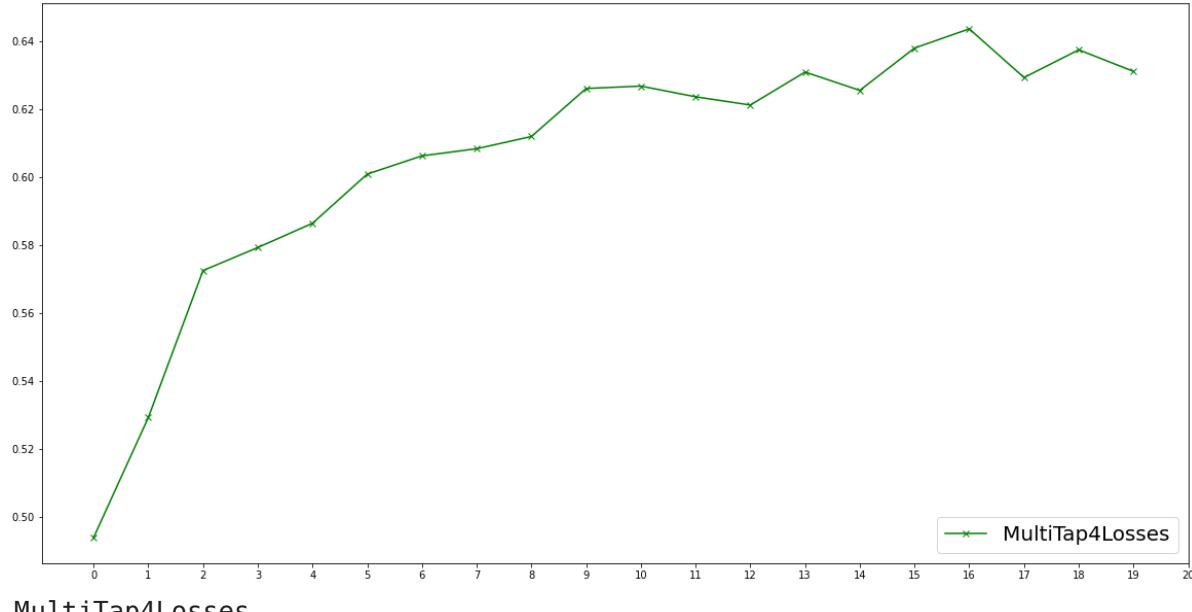
```
plt.figure(figsize=(20, 10))
MultiTap4Losses= pd.read_csv("UNET_full_9k_Feature32_MultiTap_4Losses.csv")

plt.plot(MultiTap4Losses["f1_val"],'g',marker='x',label="MultiTap4Losses")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()

maxf1 = MultiTap4Losses[MultiTap4Losses['f1_val']==max(MultiTap4Losses['f1_val'])]
print("MultiTap4Losses")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(MultiTap4Losses.tail(1))
```



```
MultiTap4Losses
max:
    Unnamed: 0      f1_val
16          17  0.643677

last epoch:
    Unnamed: 0      f1_val
19          20  0.631169
```

#

Special Channel Attention

using input image to create filters size 1 by 1

skip connection attention filter

source = UNET_full_9k_ChannelPatch_SkipOnly_f1_val.csv

In [153...]

```
plt.figure(figsize=(20, 10))
chanFilterSkip= pd.read_csv("UNET_full_9k_ChannelPatch_SkipOnly_f1_val.csv")

# plt.plot(chanFilterSkip["f1_val"],'g',marker='x',label="chanFilterSkip")
# plt.xticks(range(0,21))
# plt.legend(loc="lower right", prop={'size': 20})

# plt.show()

maxf1 = chanFilterSkip[chanFilterSkip['f1_val']==max(chanFilterSkip['f1_val'])]
print("chanFilterSkip")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(chanFilterSkip.tail(1))

chanFilterSkip
max:
    Unnamed: 0      f1_val
18           19   0.637014

last epoch:
    Unnamed: 0      f1_val
19           20   0.636468
<Figure size 1440x720 with 0 Axes>
```

decoder side attention filter

source = UNET_full_9k_ChannelPatch_UpOnly_f1_val.csv

In [156...]

```
# plt.figure(figsize=(20, 10))
chanFilterDecoder= pd.read_csv("UNET_full_9k_ChannelPatch_UpOnly_f1_val.csv")

# plt.plot(chanFilterDecoder["f1_val"],'g',marker='x',label="chanFilterDecoder")
# plt.xticks(range(0,21))
# plt.legend(loc="lower right", prop={'size': 20})

# plt.show()

maxf1 = chanFilterDecoder[chanFilterDecoder['f1_val']==max(chanFilterDecoder['f1_val'])]
print("chanFilterDecoder")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(chanFilterDecoder.tail(1))
```

```
chanFilterDecoder
max:
    Unnamed: 0      f1_val
19          20  0.641387

last epoch:
    Unnamed: 0      f1_val
19          20  0.641387
```

Encoder side attention filter

source = UNET_full_9k_ChannelPatch_DownOnly_f1_val.csv

```
In [157]: plt.figure(figsize=(20, 10))
chanFiterEncoder= pd.read_csv("UNET_full_9k_ChannelPatch_DownOnly_f1_val.csv")

# plt.plot(chanFiterEncoder["f1_val"],'g',marker='x',label="chanFiterEncoder")
# plt.xticks(range(0,21))
# plt.legend(loc="lower right", prop={'size': 20})

# plt.show()

maxf1 = chanFiterEncoder[chanFiterEncoder['f1_val']==max(chanFiterEncoder['f1_val'])]
print("chanFiterEncoder")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(chanFiterEncoder.tail(1))

chanFiterEncoder
max:
    Unnamed: 0      f1_val
18          19  0.638547

last epoch:
    Unnamed: 0      f1_val
19          20  0.630499
<Figure size 1440x720 with 0 Axes>
```

Encoder + Skip + Decoder side attention filter

source = UNET_full_9k_ChannelPatch_All3_f1_val.csv

```
In [158]: # plt.figure(figsize=(20, 10))
chanFiterAll= pd.read_csv("UNET_full_9k_ChannelPatch_All3_f1_val.csv")

# plt.plot(chanFiterAll["f1_val"],'g',marker='x',label="chanFiterAll")
# plt.xticks(range(0,21))
# plt.legend(loc="lower right", prop={'size': 20})

# plt.show()

maxf1 = chanFiterAll[chanFiterAll['f1_val']==max(chanFiterAll['f1_val'])]
print("chanFiterAll")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(chanFiterAll.tail(1))

chanFiterAll
max:
    Unnamed: 0      f1_val
19           20  0.640471

last epoch:
    Unnamed: 0      f1_val
19           20  0.640471
```

Channel Attention Compared:

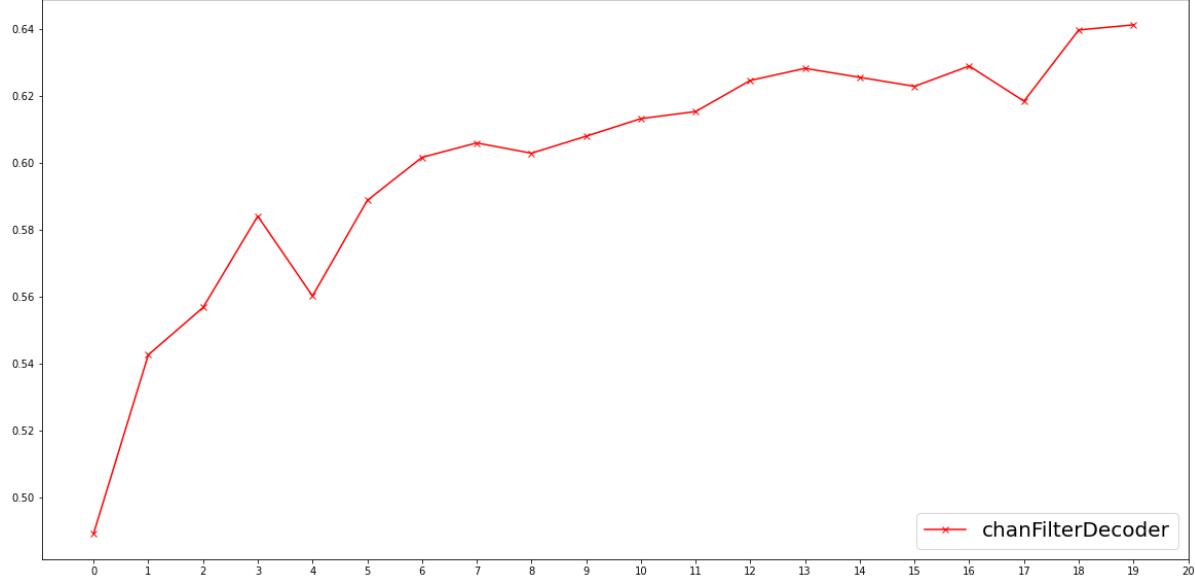
Skip vs Decoder vs Encoder vs All

```
In [159]: plt.figure(figsize=(20, 10))

# plt.plot(chanFilterSkip["f1_val"],'g',marker='x',label="chanFilterSkip")
plt.plot(chanFilterDecoder["f1_val"],'r',marker='x',label="chanFilterDecoder")
# plt.plot(chanFiterEncoder["f1_val"],'b',marker='x',label="chanFiterEncoder")
# plt.plot(chanFiterAll["f1_val"],'y',marker='x',label="chanFiterAll")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()
```



#

Comparisons of Usage of BatchNormalization

3x3 spatial attention with and without Batch Normalization

decoder side implementation

Spatial filter size is 3 overall with stride 2

```
In [13]: Patchby3 = pd.read_csv("3by3_UpOnly_f1_val_mul.csv")
maxf1 = Patchby3[Patchby3['f1_val']==max(Patchby3['f1_val'])]
print("Patchby3")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(Patchby3.tail(1))
```

```
Patchby3
max:
    step      f1_val
13     14   0.631349
```

```
last epoch:
    step      f1_val
19     20   0.62493
```

```
In [14]: Patchby3WithBN= pd.read_csv("3by3_UpOnly_withBN_f1_val_mul.csv")

maxf1 = Patchby3WithBN[Patchby3WithBN['f1_val']==max(Patchby3WithBN['f1_val'])]
print("Patchby3WithBN")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(Patchby3WithBN.tail(1))

Patchby3WithBN
max:
    Unnamed: 0      f1_val
17           18  0.635835

last epoch:
    Unnamed: 0      f1_val
19           20  0.634585
```

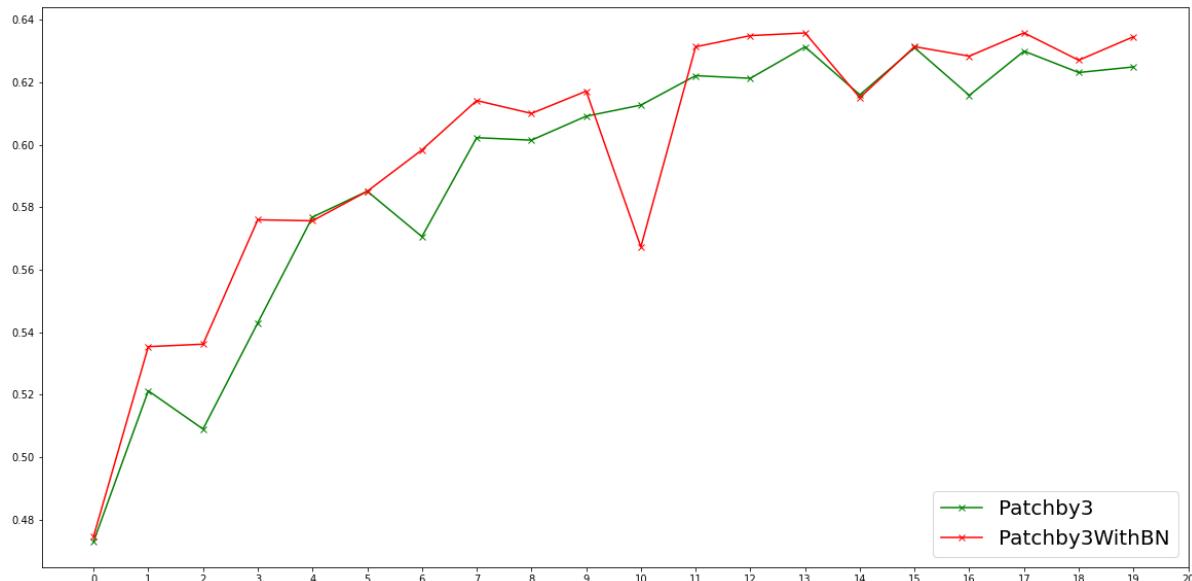
With BN vs No BN: Filter size 3

```
In [73]: plt.figure(figsize=(20, 10))

plt.plot(Patchby3["f1_val"], 'g', marker='x', label="Patchby3")
plt.plot(Patchby3WithBN["f1_val"], 'r', marker='x', label="Patchby3WithBN")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()
```



Channel Attention with and without Batch Normalization

decoder side implementation

```
In [65]: ChannelAtt = pd.read_csv("ChannelAtt_UpOnly_f1_val.csv")

maxf1 = ChannelAtt[ChannelAtt['f1_val']==max(ChannelAtt['f1_val'])]
print("ChannelAtt")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(ChannelAtt.tail(1))
```

```
ChannelAtt
max:
    Unnamed: 0      f1_val
19           20  0.641387
```

```
last epoch:
    Unnamed: 0      f1_val
19           20  0.641387
```

```
In [66]: ChannelAttWithBN = pd.read_csv("ChannelAtt_UpOnly_withBN_f1_val.csv")
```

```
maxf1 = ChannelAttWithBN[ChannelAttWithBN['f1_val']==max(ChannelAttWithBN['f1_val'])]
print("ChannelAttWithBN")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(ChannelAttWithBN.tail(1))
```

```
ChannelAttWithBN
max:
    Unnamed: 0      f1_val
18           19  0.639358
```

```
last epoch:
    Unnamed: 0      f1_val
19           20  0.638504
```

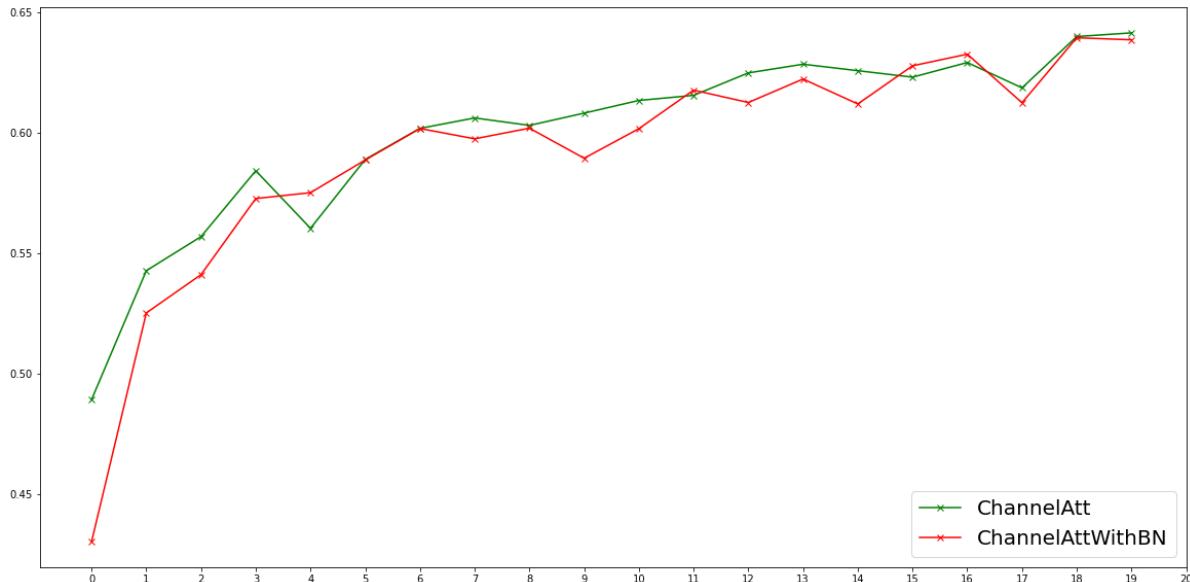
Channel Attention NoBN (green) vs BN (red)

```
In [67]: plt.figure(figsize=(20, 10))

plt.plot(ChannelAtt["f1_val"], 'g', marker='x', label="ChannelAtt")
plt.plot(ChannelAttWithBN["f1_val"], 'r', marker='x', label="ChannelAttWithBN")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()
```



#

BN on spatial and channel attention experiment

Implemented in skip, decoder and encoder

Spatial filter size is 3 overall with stride 2

```
In [68]: SpatialAndChannelOnly = pd.read_csv("3by3FilterPlusChanAttentionNoBN_f1_val_
maxf1 = SpatialAndChannelOnly[SpatialAndChannelOnly['f1_val']==max(SpatialA
print("SpatialAndChannelOnly")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(SpatialAndChannelOnly.tail(1))
```

```
SpatialAndChannelOnly
max:
      Unnamed: 0    f1_val
16           16  0.56445

last epoch:
      Unnamed: 0    f1_val
19           19  0.509556
```

```
In [69]: BNonspatialAndChannel = pd.read_csv("3by3BNonspatialAndChannel_f1_val_mul.csv")

maxf1 = BNonspatialAndChannel[BNonspatialAndChannel['f1_val']==max(BNonspatialAndChannel['f1_val'])]
print("BNonspatialAndChannel")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(BNonspatialAndChannel.tail(1))
```

```
BNonspatialAndChannel
max:
    Unnamed: 0      f1_val
18          18  0.629485

last epoch:
    Unnamed: 0      f1_val
19          19  0.618202
```

```
In [70]: BNonspatialButNotChannel = pd.read_csv("3by3withChannelAttentionBNonspatial.csv")

maxf1 = BNonspatialButNotChannel[BNonspatialButNotChannel['f1_val']==max(BNonspatialButNotChannel['f1_val'])]
print("BNonspatialButNotChannel")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(BNonspatialButNotAttention.tail(1))
```

```
BNonspatialButNotChannel
max:
    Unnamed: 0      f1_val
14          14  0.614083

last epoch:
    Unnamed: 0      f1_val
19          19  0.593635
```

```
In [71]: BNonChannelNotSpatial = pd.read_csv("3by3FilterPlusChanAttentionBNonChannel.csv")

maxf1 = BNonChannelNotSpatial[BNonChannelNotSpatial['f1_val']==max(BNonChannelNotSpatial['f1_val'])]
print("BNonChannelNotSpatial")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(BNonChannelNotSpatial.tail(1))
```

```
BNonChannelNotSpatial
max:
    Unnamed: 0      f1_val
16          16  0.631086

last epoch:
    Unnamed: 0      f1_val
19          19  0.619472
```

BN on Spatial (red) vs BN on Channel (green) vs BN on both (blue) Channel and spatial vs NO BN

In []:

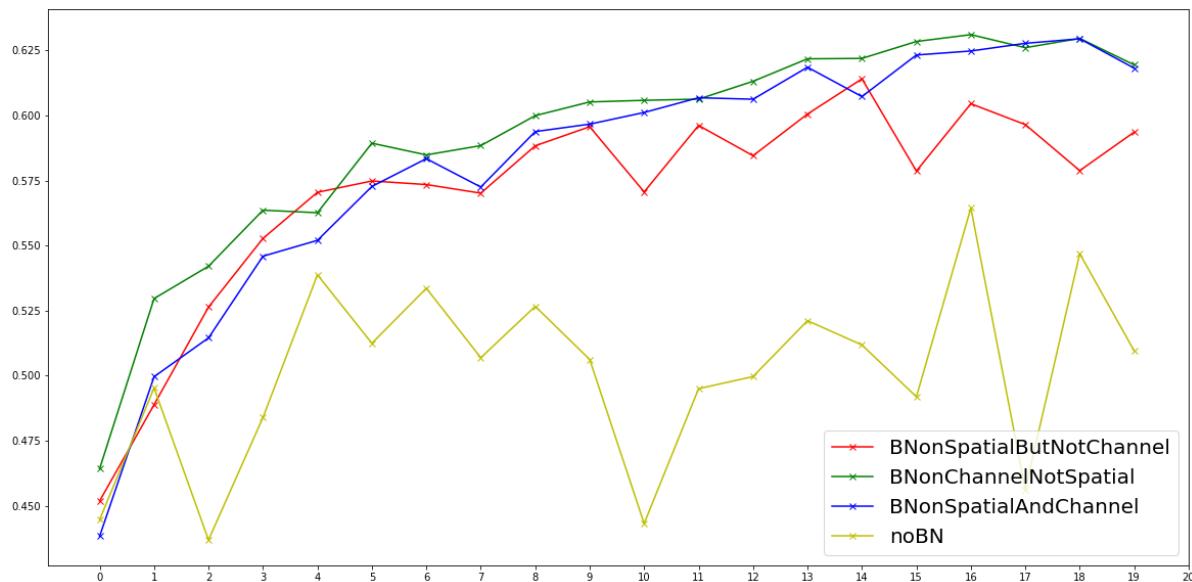
In [72]:

```
plt.figure(figsize=(20, 10))

plt.plot(BNonSpatialButNotChannel["f1_val"], 'r', marker='x', label="BNonSpatialButNotChannel")
plt.plot(BNonChannelNotSpatial["f1_val"], 'g', marker='x', label="BNonChannelNotSpatial")
plt.plot(BNonSpatialAndChannel["f1_val"], 'b', marker='x', label="BNonSpatialAndChannel")
plt.plot(SpatialAndAttentionOnly["f1_val"], 'y', marker='x', label="noBN")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()
```



Experiment where Channel Attention applied first then spatial

In [76]:

```
BNonSpatialAndChannelReverse = pd.read_csv("3by3BNonSpatialAndChannel-Revers
maxf1 = BNonSpatialAndChannelReverse[BNonSpatialAndChannelReverse['f1_val']=
print("BNonSpatialAndChannelReverse")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(BNonSpatialAndChannelReverse.tail(1))
```

BNonSpatialAndChannelReverse

max:
 Unnamed: 0 f1_val
16 16 0.627673

last epoch:
 Unnamed: 0 f1_val
19 19 0.599747

ChannelToSpatial vs SpatialToChannel

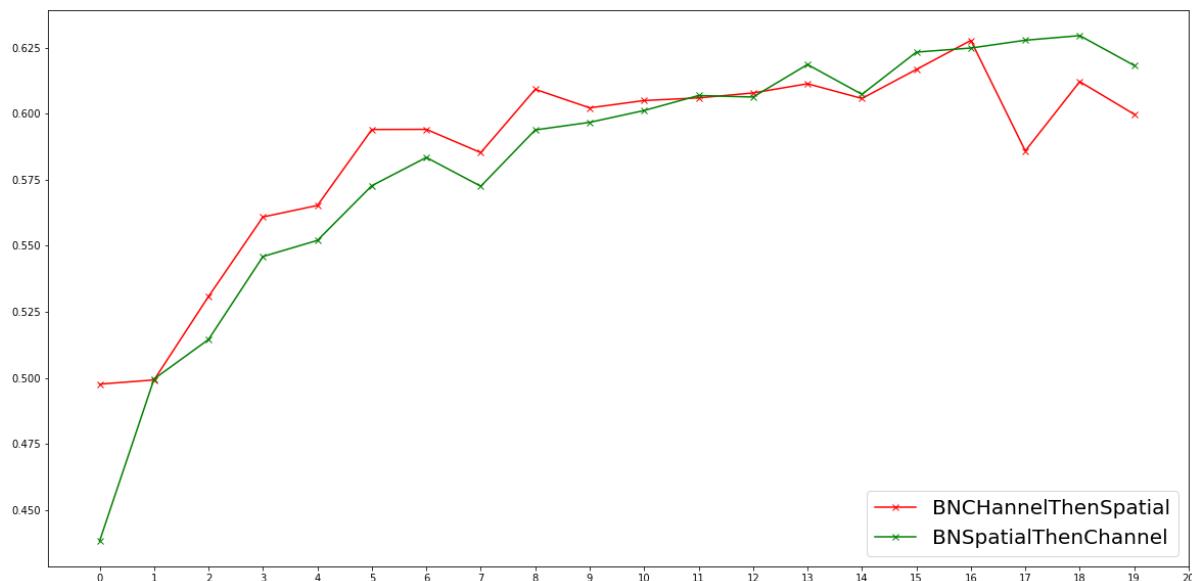
BN used

```
In [83]: plt.figure(figsize=(20, 10))

plt.plot(BNonSpatialAndChannelReverse["f1_val"], 'r', marker='x', label="BNChannelThenSpatial")
plt.plot(BNonSpatialAndChannel["f1_val"], 'g', marker='x', label="BNSpatialThenChannel")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()
```



```
In [81]: SpatialAndChannelReverseNoBN = pd.read_csv("3by3FilterPlusChanAttentionNoBN.csv")
maxf1 = SpatialAndChannelReverseNoBN[SpatialAndChannelReverseNoBN['f1_val'].index[-1]]
print("SpatialAndChannelReverseNoBN")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(SpatialAndChannelReverseNoBN.tail(1))
```

SpatialAndChannelReverseNoBN

```
max:
      Unnamed: 0    f1_val
12           12  0.593879
```

```
last epoch:
      Unnamed: 0    f1_val
19           19  0.553558
```

ChannelToSpatial vs SpatialToChannel

NO BN

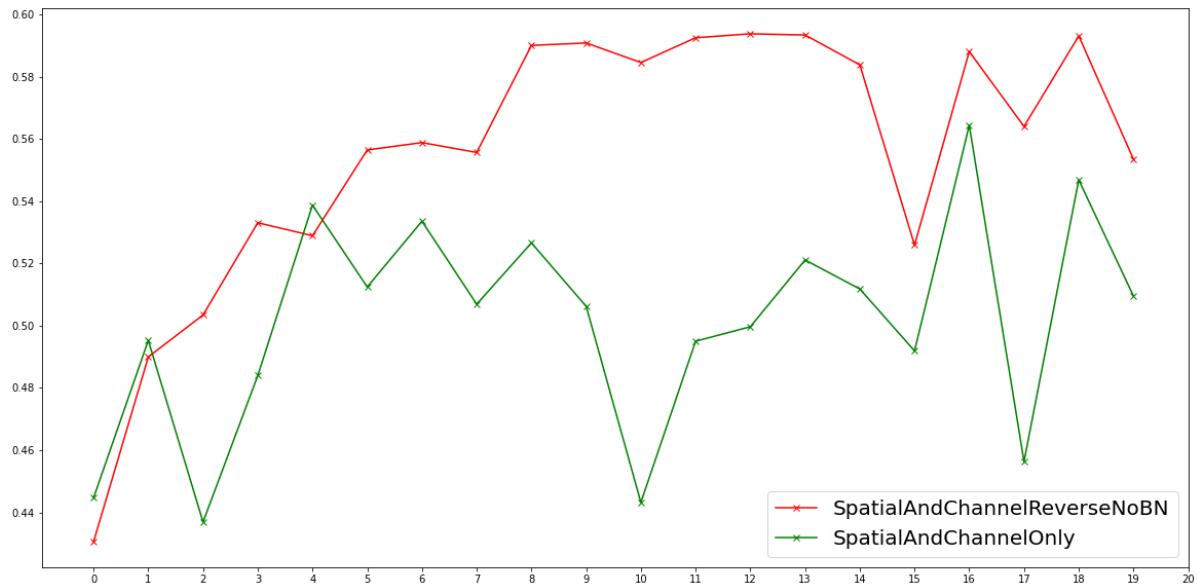
```
In [82]: SpatialAndChannelOnly
```

```
plt.figure(figsize=(20, 10))

plt.plot(SpatialAndChannelReverseNoBN["f1_val"], 'r', marker='x', label="SpatialAndChannelReverseNoBN")
plt.plot(SpatialAndChannelOnly["f1_val"], 'g', marker='x', label="SpatialAndChannelOnly")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()
```



```
#
```

Linear to combine spatial attention instead of weights

```
In [84]:
```

```
LinearSpatialCombo = pd.read_csv("Combined_3579_StrideHalfFilterAggregates_f1_val.csv")
maxf1 = LinearSpatialCombo[LinearSpatialCombo['f1_val']==max(LinearSpatialCombo['f1_val'])]
print("LinearSpatialCombo")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(LinearSpatialCombo.tail(1))
```

```
LinearSpatialCombo
```

```
max:
```

```
    Unnamed: 0      f1_val
9             9  0.481994
```

```
last epoch:
```

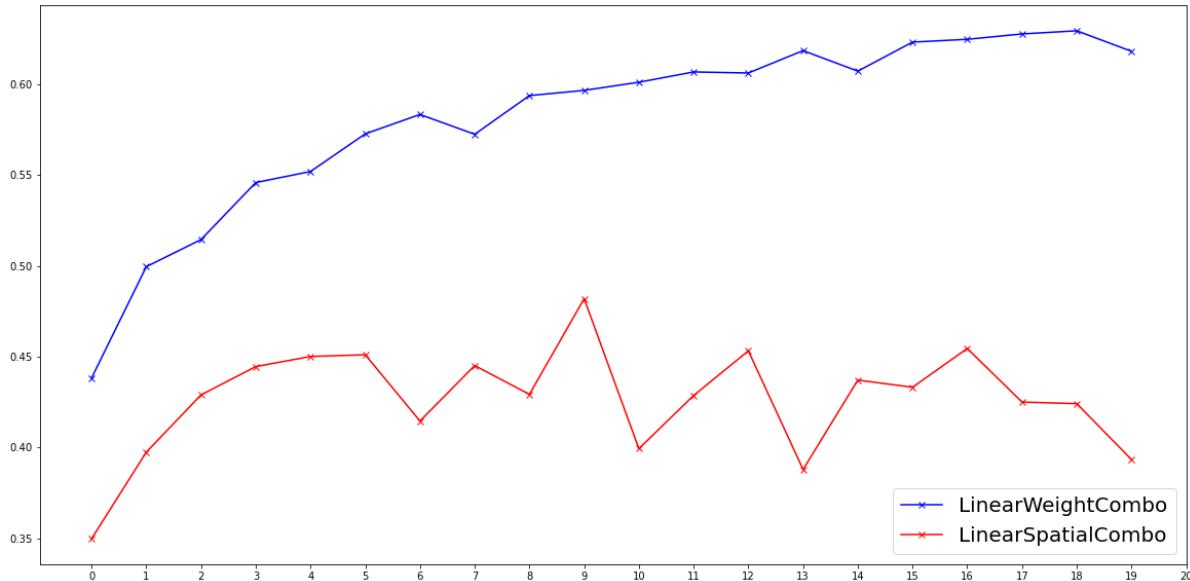
```
    Unnamed: 0      f1_val
19            19  0.393471
```

```
In [86]: plt.figure(figsize=(20, 10))
```

```
plt.plot(BNonSpatialAndChannel["f1_val"], 'b', marker='x', label="LinearWeightC")  
plt.plot(LinearSpatialCombo["f1_val"], 'r', marker='x', label="LinearSpatialCom")
```

```
plt.xticks(range(0,21))  
plt.legend(loc="lower right", prop={"size": 20})
```

```
plt.show()
```



Only Attention

```
In [87]: OnlyAttention = pd.read_csv("Combined_3579_PlusChannel_NoBasicBlock_f1_val_n  
maxf1 = OnlyAttention[OnlyAttention['f1_val']==max(OnlyAttention['f1_val'])]  
print("OnlyAttention")  
print("max:")  
print(maxf1)  
print()  
print("last epoch:")  
print(OnlyAttention.tail(1))
```

```
OnlyAttention  
max:  
    Unnamed: 0      f1_val  
17          17  0.477655
```

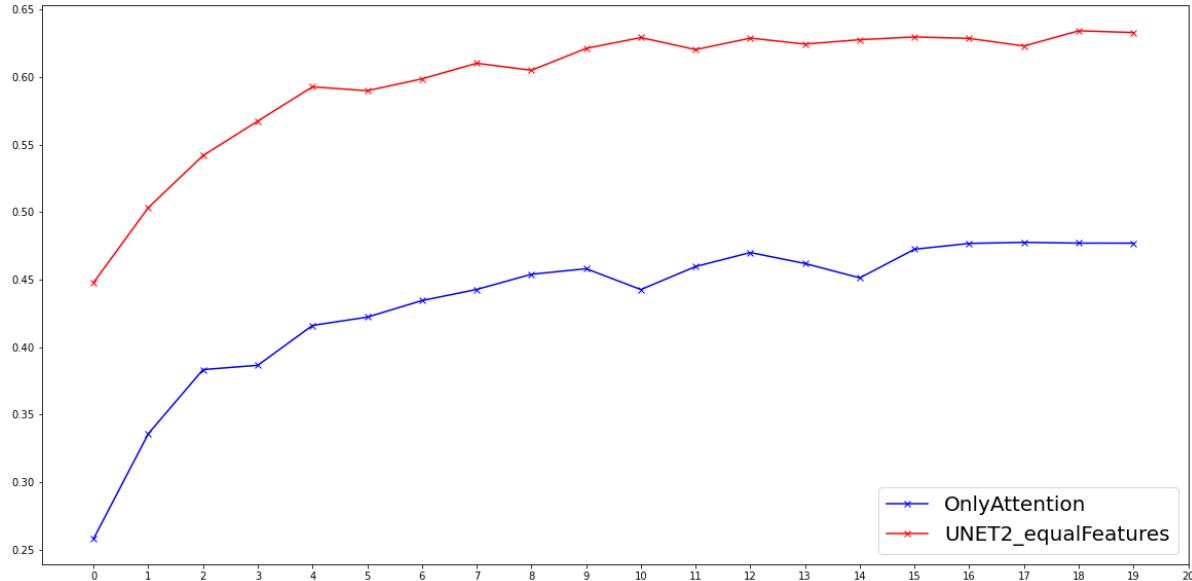
```
last epoch:  
    Unnamed: 0      f1_val  
19          19  0.47706
```

```
In [92]: plt.figure(figsize=(20, 10))
```

```
plt.plot(OnlyAttention["f1_val"],'b',marker='x',label="OnlyAttention")
plt.plot(history_Feature32_full["F1_val"],'r',marker='x',label="UNET2_equalFeatures")
```

```
plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})
```

```
plt.show()
```



Variational UNET two methods

a) Using Convolution

b) Using Flatten and Linear

Single UNET with no bottleneck

a) using convolution

```
In [5]: UNetConvolution = pd.read_csv("Unet2_convVAE.csv")
maxf1 = UNetConvolution[UNetConvolution['f1_val']==max(UNetConvolution['f1_val'])]
print("UNetConvolution")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(UNetConvolution.tail(1))
```

```
UNetConvolution
max:
    Unnamed: 0      f1_val
19          19  0.63623

last epoch:
    Unnamed: 0      f1_val
19          19  0.63623
```

SingleUNET with no bottleneck

b) using Linear

```
In [35]: UNetLinear = pd.read_csv("Unet2_VAE_Flatten_f1_val.csv")
maxf1 = UNetLinear[UNetLinear['f1_val']==max(UNetLinear['f1_val'])]
print("UNetFlatten")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(UNetFlatten.tail(1))

UNetFlatten
max:
    Unnamed: 0      f1_val
17          17  0.628346

last epoch:
    Unnamed: 0      f1_val
19          19  0.624418
```

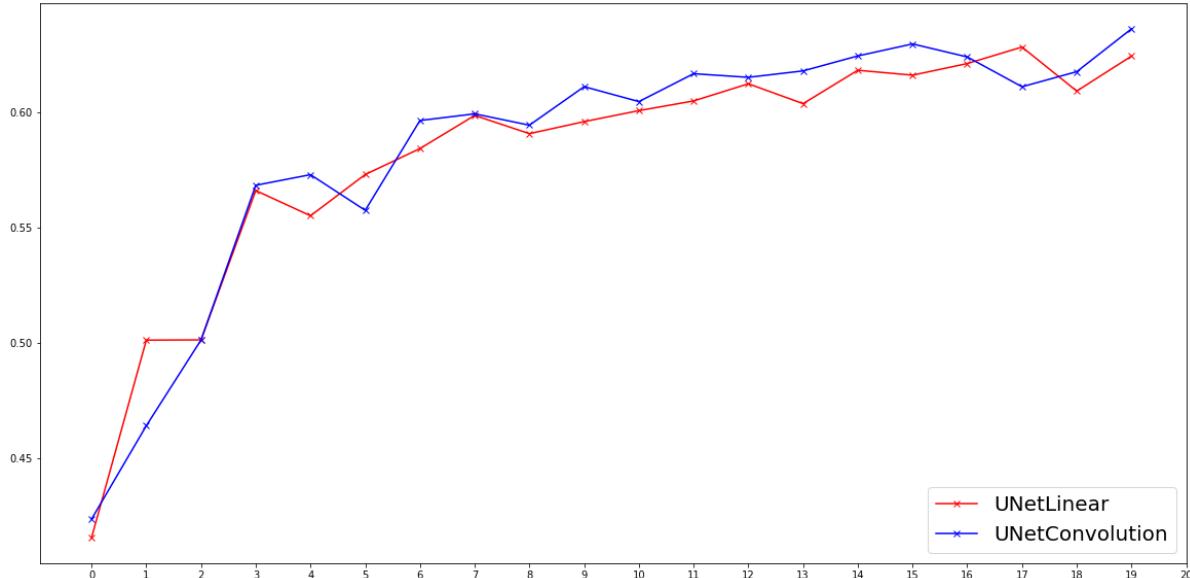
Linear vs Conv No Bottleneck

```
In [38]: plt.figure(figsize=(20, 10))

plt.plot(UNetLinear["f1_val"], 'r', marker='x', label="UNetLinear")
plt.plot(UNetConvolution["f1_val"], 'b', marker='x', label="UNetConvolution")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()
```



Generate xhat then retrain for yhat

a) Convolution

```
In [39]: UNetConvolutionXhat = pd.read_csv("Unet2_ConVAE_NoSkipConnection_retrainWith_f1_val")
maxf1 = UNetConvolutionXhat[UNetConvolutionXhat['f1_val']==max(UNetConvolutionXhat['f1_val'])]
print("UNetConvolutionXhat")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(UNetConvolutionXhat.tail(1))

UNetConvolutionXhat
max:
      Unnamed: 0      f1_val
17           17  0.642876

last epoch:
      Unnamed: 0      f1_val
19           19  0.637784
```

Generate xhat then retrain for yhat

b) Linear Method

```
In [40]: UNetLinearXhat = pd.read_csv("Unet2_VAE_NoSkipConnection_retrainWith_f1_val")
maxf1 = UNetLinearXhat[UNetLinearXhat['f1_val']==max(UNetLinearXhat['f1_val'])]
print("UNetLinearXhat")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(UNetLinearXhat.tail(1))
```

```
UNetLinearXhat
max:
    Unnamed: 0      f1_val
16          16  0.641835

last epoch:
    Unnamed: 0      f1_val
19          19  0.636959
```

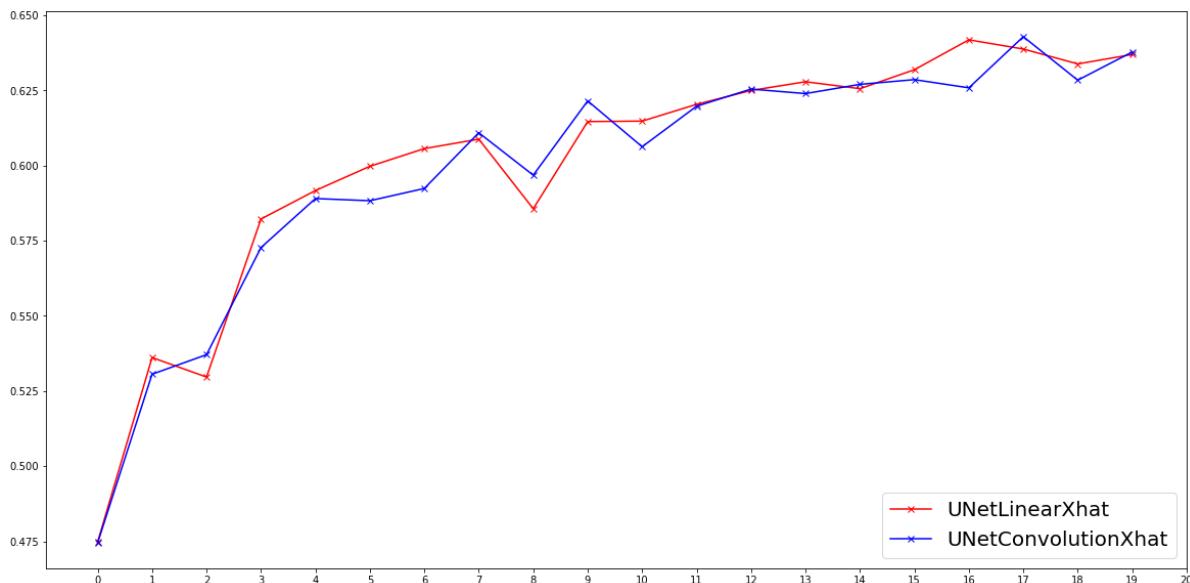
Linear vs Conv repeated training

```
In [41]: plt.figure(figsize=(20, 10))

plt.plot(UNetLinearXhat["f1_val"], 'r', marker='x', label="UNetLinearXhat")
plt.plot(UNetConvolutionXhat["f1_val"], 'b', marker='x', label="UNetConvolutionXhat")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()
```



VAE in skip connections

a) convolution

```
In [42]: UNetConvolutionSkip = pd.read_csv("Unet2_SkipConVAE_f1_val.csv")
maxf1 = UNetConvolutionSkip[UNetConvolutionSkip['f1_val']==max(UNetConvolutionSkip['f1_val'])]
print("UNetConvolutionSkip")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(UNetConvolutionSkip.tail(1))
```

```
UNetConvolutionSkip
max:
    Unnamed: 0      f1_val
18          18  0.32256

last epoch:
    Unnamed: 0      f1_val
19          19  0.318902
```

VAE in skip connections

b) Linear

```
In [43]: UNetLinearSkip = pd.read_csv("Unet2_SkipVAE_Linear_f1_val.csv")
maxf1 = UNetLinearSkip[UNetLinearSkip['f1_val']==max(UNetLinearSkip['f1_val'])]
print("UNetLinearSkip")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(UNetLinearSkip.tail(1))

UNetLinearSkip
max:
    Unnamed: 0      f1_val
19          19  0.482236

last epoch:
    Unnamed: 0      f1_val
19          19  0.482236
```

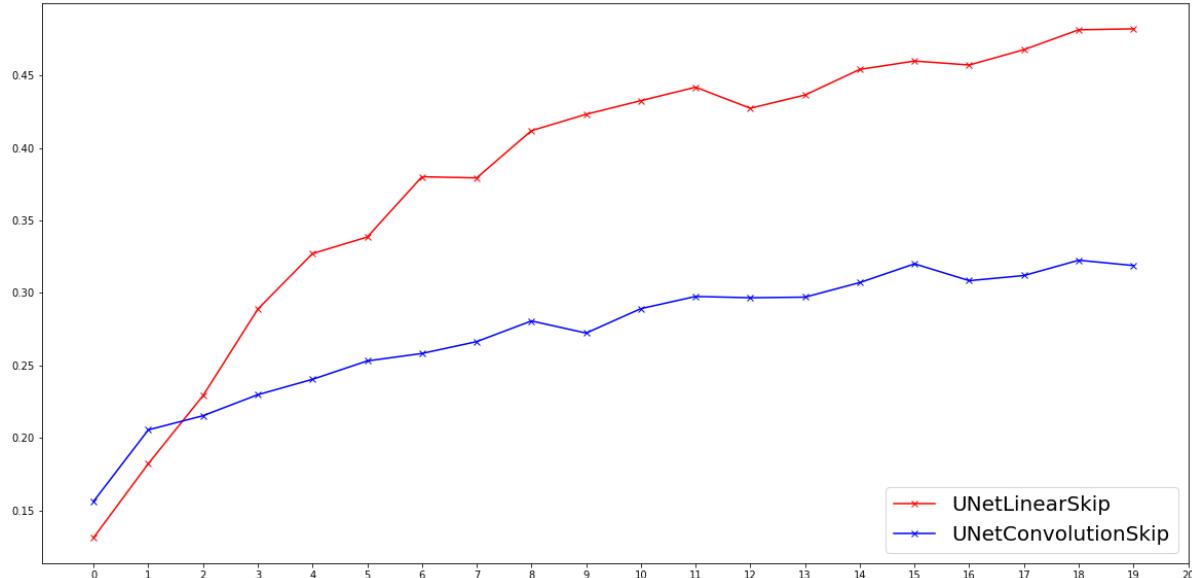
Linear vs convolution skip connections

```
In [44]: plt.figure(figsize=(20, 10))

plt.plot(UNetLinearSkip["f1_val"],'r',marker='x',label="UNetLinearSkip")
plt.plot(UNetConvolutionSkip["f1_val"],'b',marker='x',label="UNetConvolutionSkip")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()
```



VAE with 2 Decoders

a) Convolution

```
In [45]: UNetConvolution2Decoders = pd.read_csv("ConVAE_2Decoders_f1_val.csv")
maxf1 = UNetConvolution2Decoders[UNetConvolution2Decoders['f1_val']==max(UNetConvolution2Decoders['f1_val'])]
print("UNetConvolution2Decoders")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(UNetConvolution2Decoders.tail(1))
```

```
UNetConvolution2Decoders
max:
    Unnamed: 0      f1_val
19           19  0.60086

last epoch:
    Unnamed: 0      f1_val
19           19  0.60086
```

```
In [46]: UNetLinear2Decoders = pd.read_csv("LinearVAE_2Decoders_f1_val.csv")
maxf1 = UNetLinear2Decoders[UNetLinear2Decoders['f1_val']==max(UNetLinear2Decoders['f1_val'])]
print("UNetLinear2Decoders")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(UNetLinear2Decoders.tail(1))
```

```
UNetLinear2Decoders
max:
    Unnamed: 0      f1_val
18          18  0.627016

last epoch:
    Unnamed: 0      f1_val
19          19  0.623513
```

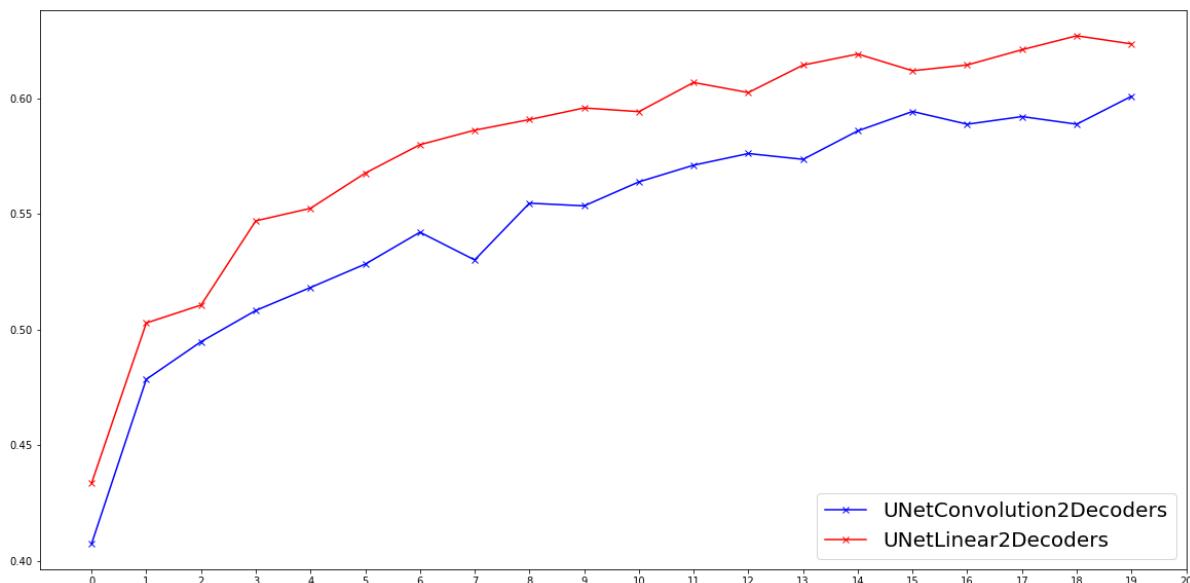
Linear vs convolution 2 Decoders

```
In [47]: plt.figure(figsize=(20, 10))
```

```
plt.plot(UNetConvolution2Decoders["f1_val"], 'b', marker='x', label="UNetConvolution2Decoders")
plt.plot(UNetLinear2Decoders["f1_val"], 'r', marker='x', label="UNetLinear2Decoders")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()
```



VAE decoder following decoder with KLD loss only and Freezing encoder layers

a) convolution

```
In [51]: ConvDecoder2DecoderFreezeKL = pd.read_csv("Unet2_ConVAE_DecoderFollowingDecoderFreezeKL.csv")
maxf1 = ConvDecoder2DecoderFreezeKL[ConvDecoder2DecoderFreezeKL['f1_val']==max(ConvDecoder2DecoderFreezeKL['f1_val'])]
print("ConvDecoder2DecoderFreezeKL")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(ConvDecoder2DecoderFreezeKL.tail(1))
```

```
ConvDecoder2DecoderFreezeKL
max:
    Unnamed: 0      f1_val
17          17  0.545808

last epoch:
    Unnamed: 0      f1_val
19          19  0.545627
```

b) Linear

```
In [52]: LinDecoder2DecoderFreezeKL= pd.read_csv("Unet2_VAE__DecoderFollowingDecoder_
maxf1 = LinDecoder2DecoderFreezeKL[LinDecoder2DecoderFreezeKL['f1_val']==max]
print("LinDecoder2DecoderFreezeKL")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(LinDecoder2DecoderFreezeKL.tail(1))
```

```
LinDecoder2DecoderFreezeKL
max:
    Unnamed: 0      f1_val
16          16  0.540976

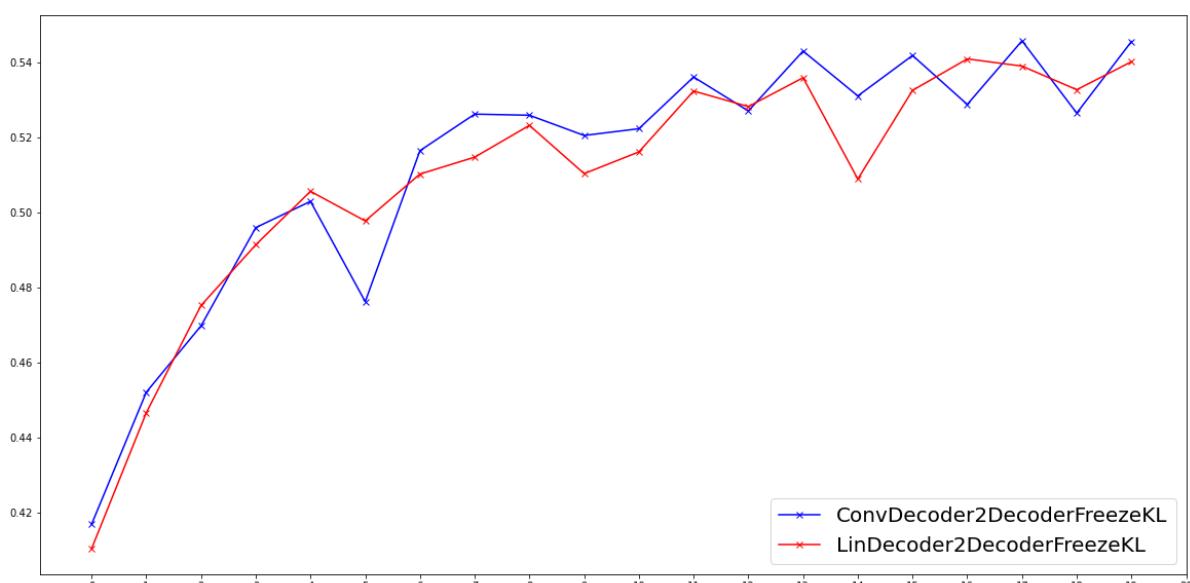
last epoch:
    Unnamed: 0      f1_val
19          19  0.540233
```

```
In [63]: plt.figure(figsize=(20, 10))

plt.plot(ConvDecoder2DecoderFreezeKL["f1_val"], 'b', marker='x', label="ConvDecoder2DecoderFreezeKL")
plt.plot(LinDecoder2DecoderFreezeKL["f1_val"], 'r', marker='x', label="LinDecoder2DecoderFreezeKL")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()
```



VAE decoder following decoder with KLD loss Plus Our Loss and Freezing encoder layers

a) convolution

In [54]:

```
ConvDecoder2DecoderFreezeOnly = pd.read_csv("Unet2_ConVAE_DecoderFollowingDecoderFreezeOnly.csv")
maxf1 = ConvDecoder2DecoderFreezeOnly[ConvDecoder2DecoderFreezeOnly['f1_val'].idxmax()]
print("ConvDecoder2DecoderFreezeOnly")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(ConvDecoder2DecoderFreezeOnly.tail(1))
```

ConvDecoder2DecoderFreezeOnly

max:
 Unnamed: 0 f1_val
19 19 0.54348

last epoch:

 Unnamed: 0 f1_val
19 19 0.54348

b) Linear

In [56]:

```
LinDecoder2DecoderFreezeOnly = pd.read_csv("Unet2_VAE__DecoderFollowingDecoderFreezeOnly.csv")
maxf1 = LinDecoder2DecoderFreezeOnly[LinDecoder2DecoderFreezeOnly['f1_val'].idxmax()]
print("LinDecoder2DecoderFreezeOnly")
print("max:")

print(maxf1)
print()
print("last epoch:")
print(LinDecoder2DecoderFreezeOnly.tail(1))
```

LinDecoder2DecoderFreezeOnly

max:
 Unnamed: 0 f1_val
19 19 0.540118

last epoch:

 Unnamed: 0 f1_val
19 19 0.540118

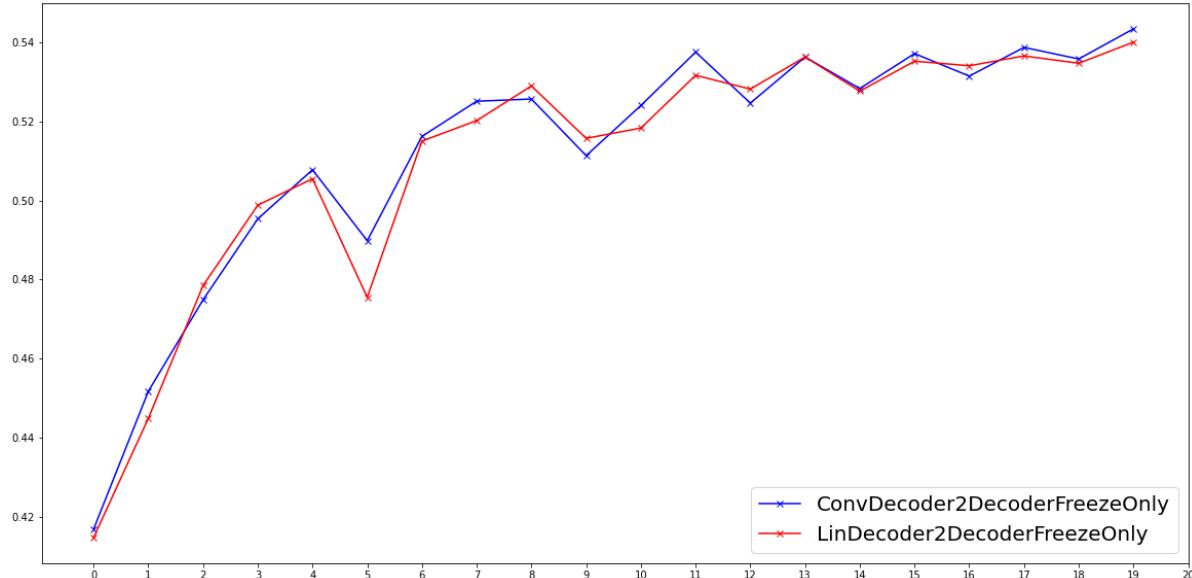
In [64]:

```
plt.figure(figsize=(20, 10))

plt.plot(ConvDecoder2DecoderFreezeOnly["f1_val"], 'b', marker='x', label="ConvDecoder2DecoderFreezeOnly")
plt.plot(LinDecoder2DecoderFreezeOnly["f1_val"], 'r', marker='x', label="LinDecoder2DecoderFreezeOnly")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()
```



VAE decoder following decoder with KLD loss Plus Our Loss and NO Freezing

a) convolution

```
In [61]: ConvDecoder2DecoderNoFreeze = pd.read_csv("Unet2_ConVAE_DecoderFollowingDecoderNoFreeze.csv")
maxf1 = ConvDecoder2DecoderNoFreeze[ConvDecoder2DecoderNoFreeze['f1_val']==max]
print("ConvDecoder2DecoderNoFreeze")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(ConvDecoder2DecoderNoFreeze.tail(1))
```

```
ConvDecoder2DecoderNoFreeze
max:
      Unnamed: 0      f1_val
16           16  0.640325

last epoch:
      Unnamed: 0      f1_val
19           19  0.637349
```

b) Linear

```
In [62]: LinDecoder2DecoderNoFreeze = pd.read_csv("Unet2_VAE_DecoderFollowingDecoderNoFreeze.csv")
maxf1 = LinDecoder2DecoderNoFreeze[LinDecoder2DecoderNoFreeze['f1_val']==max]
print("LinDecoder2DecoderNoFreeze")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(LinDecoder2DecoderNoFreeze.tail(1))
```

LinDecoder2DecoderNoFreeze

max:

	Unnamed: 0	f1_val
16	16	0.64184

last epoch:

	Unnamed: 0	f1_val
19	19	0.640717

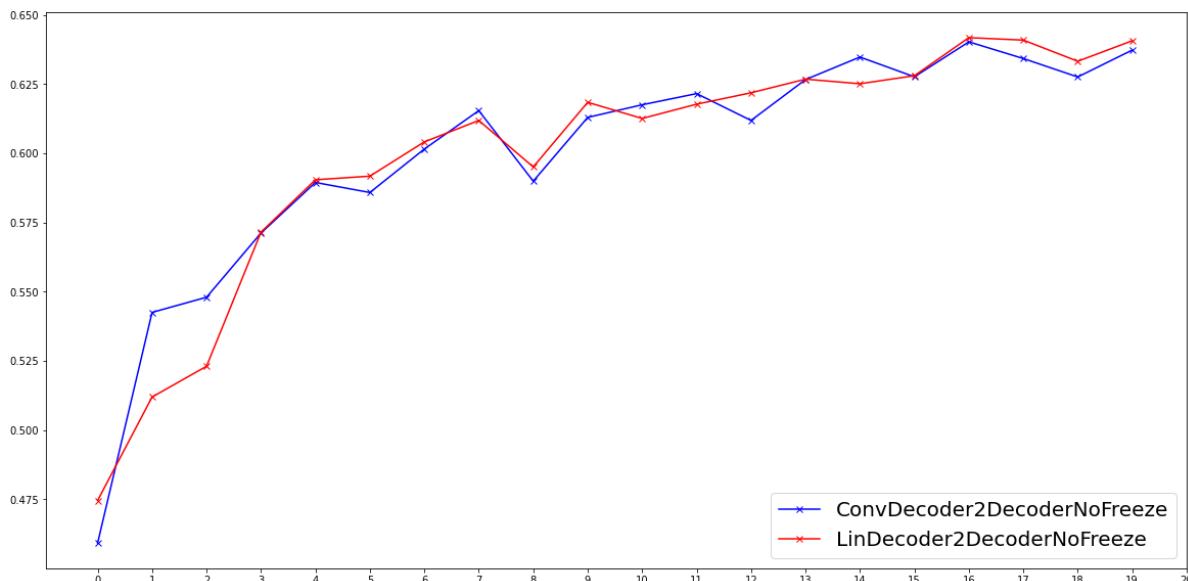
In [65]:

```
plt.figure(figsize=(20, 10))

plt.plot(ConvDecoder2DecoderNoFreeze["f1_val"], 'b', marker='x', label="ConvDecoder2DecoderNoFreeze")
plt.plot(LinDecoder2DecoderNoFreeze["f1_val"], 'r', marker='x', label="LinDecoder2DecoderNoFreeze")

plt.xticks(range(0,21))
plt.legend(loc="lower right", prop={'size': 20})

plt.show()
```



Linear mu sigma calculation nonSigmoid Twin Decoders VAE plus Normal Decoder 3 Channels

In [4]:

LinearVAE_2Decoders_nonSigmoid3Channels_f1_val.csv

In [10]:

```
LinTwinDecoder3ChannelNonSig = pd.read_csv("LinearVAE_2Decoders_nonSigmoid3Channels_f1_val.csv")
maxf1 = LinTwinDecoder3ChannelNonSig[LinTwinDecoder3ChannelNonSig['f1_val'].index[-1]]
print("LinTwinDecoder3ChannelNonSig")
print("max:")
print(maxf1)
print()
print("last epoch:")
print(LinTwinDecoder3ChannelNonSig.tail(1))
```

```
LinTwinDecoder3ChannelNonSig
```

```
max:
```

```
    Unnamed: 0      f1_val  
17          17  0.627798
```

```
last epoch:
```

```
    Unnamed: 0      f1_val  
19          19  0.614941
```

```
In [11]:
```

```
LinTwinDecoderNonSig = pd.read_csv("LinearVAE_2Decoders_SingleChannel_f1_val.csv")  
maxf1 = LinTwinDecoderNonSig[LinTwinDecoderNonSig['f1_val']==max(LinTwinDecoderNonSig['f1_val'])]  
print("LinTwinDecoderNonSig")  
print("max:")  
print(maxf1)  
print()  
print("last epoch:")  
print(LinTwinDecoderNonSig.tail(1))
```

```
LinTwinDecoderNonSig
```

```
max:
```

```
    Unnamed: 0      f1_val  
18          18  0.626162
```

```
last epoch:
```

```
    Unnamed: 0      f1_val  
19          19  0.614363
```

```
In [13]:
```

```
LinTwinDecoderSig = pd.read_csv("LinearVAE_2Decoders_MSE_1ChannelSigmoid_f1_val.csv")  
maxf1 = LinTwinDecoderSig[LinTwinDecoderSig['f1_val']==max(LinTwinDecoderSig['f1_val'])]  
print("LinTwinDecoderSig")  
print("max:")  
print(maxf1)  
print()  
print("last epoch:")  
print(LinTwinDecoderSig.tail(1))
```

```
LinTwinDecoderSig
```

```
max:
```

```
    Unnamed: 0      f1_val  
17          17  0.629113
```

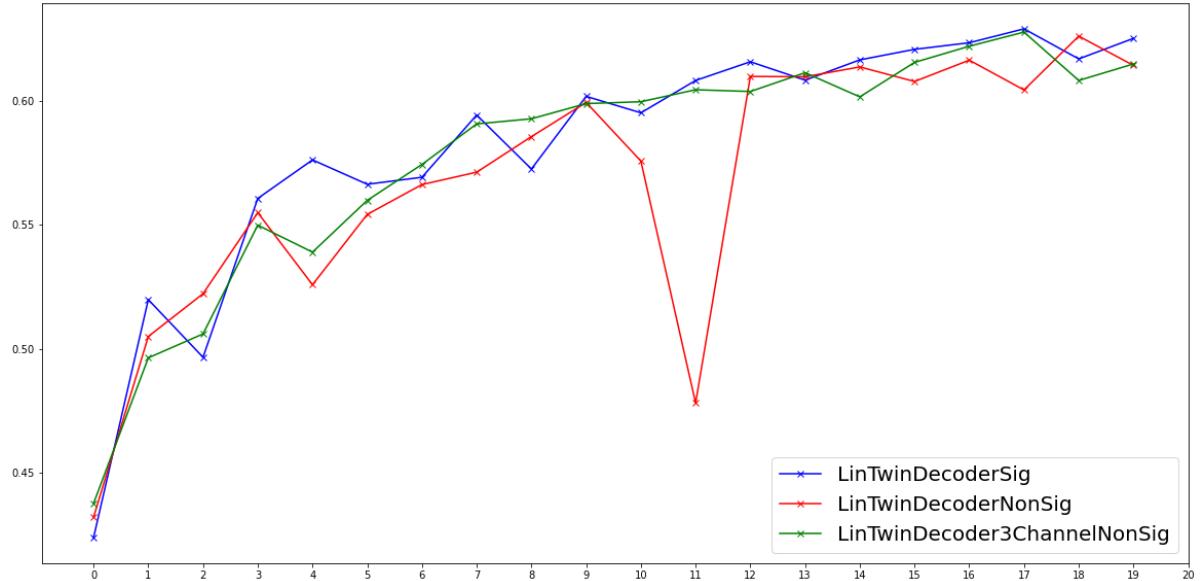
```
last epoch:
```

```
    Unnamed: 0      f1_val  
19          19  0.62526
```

```
In [14]:
```

```
plt.figure(figsize=(20, 10))
```

```
plt.plot(LinTwinDecoderSig["f1_val"], 'b', marker='x', label="LinTwinDecoderSig")  
plt.plot(LinTwinDecoderNonSig["f1_val"], 'r', marker='x', label="LinTwinDecoderNonSig")  
plt.plot(LinTwinDecoder3ChannelNonSig["f1_val"], 'g', marker='x', label="LinTwinDecoder3ChannelNonSig")  
  
plt.xticks(range(0,21))  
plt.legend(loc="lower right", prop={'size': 20})  
  
plt.show()
```



In []:

In []:

In []:

In []: