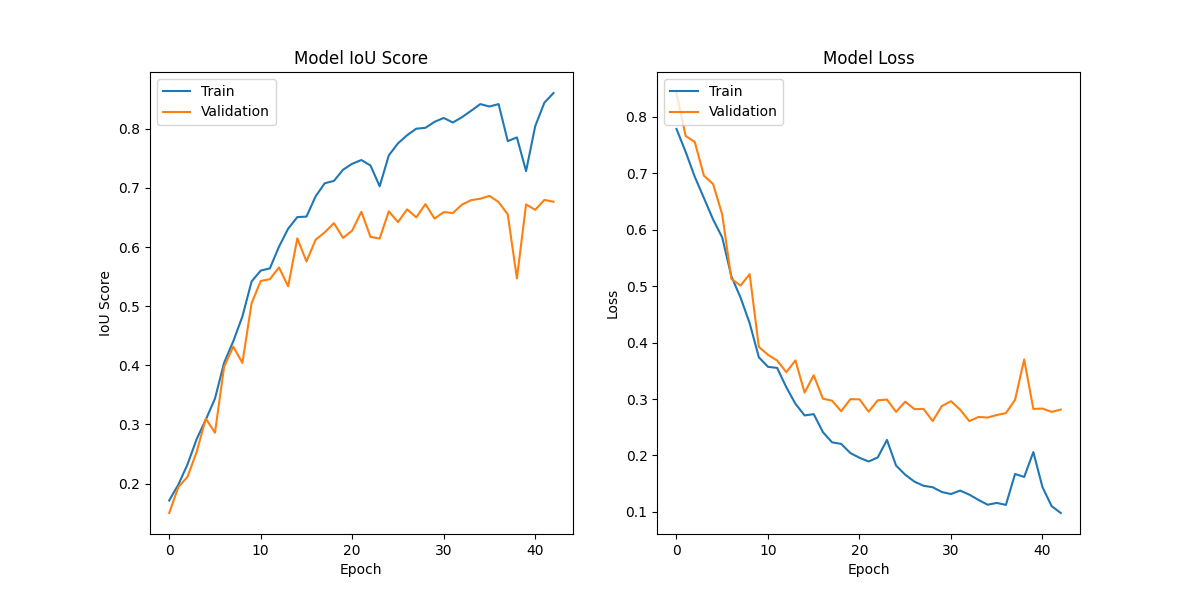
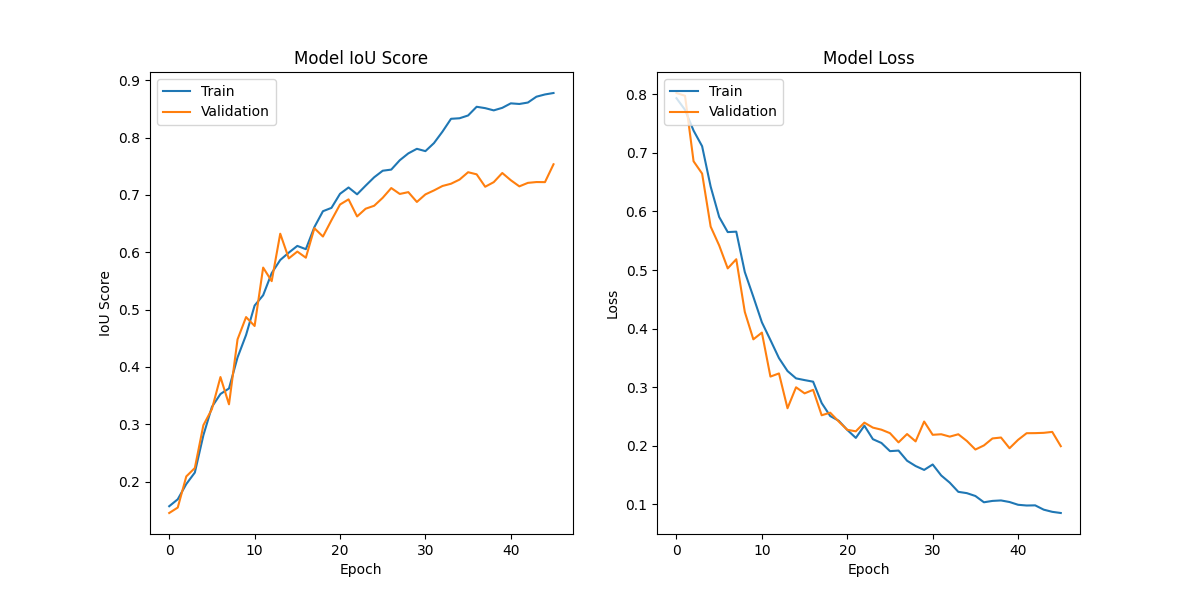
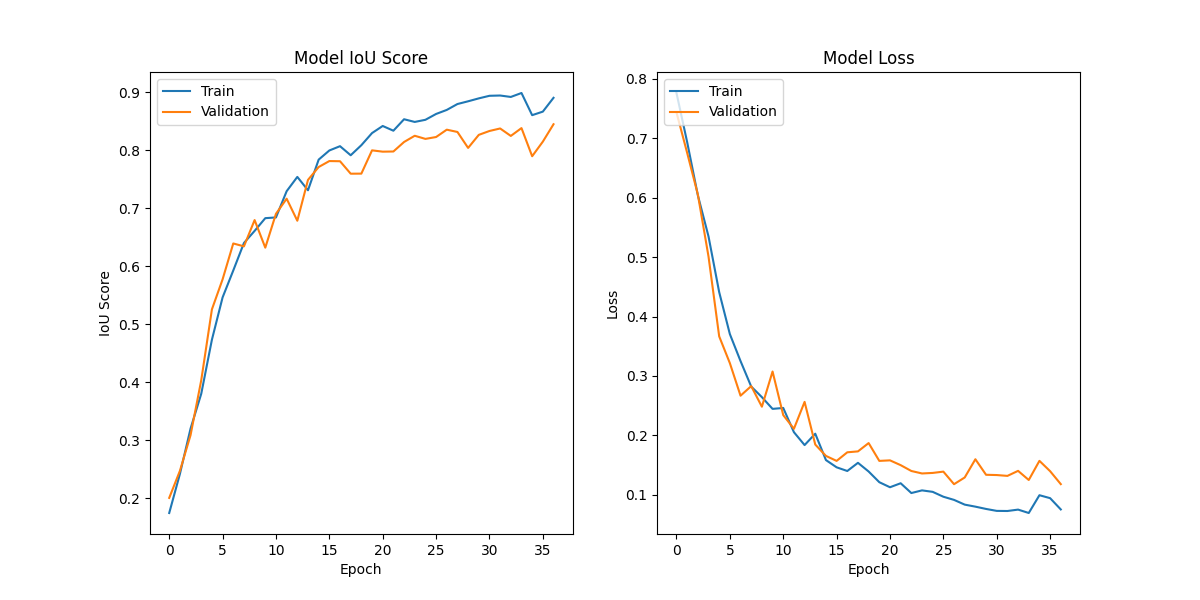
Patched images to 200x200  
  
loss: 0.0614 - iou\_score: 0.9101 - val\_loss: 0.3040 - val\_iou\_score: 0.6919  
  
  
the val iou is poor.  
  
I want to boost it.  
imma get a first 16 filter layer line  


31ms/step - loss: 0.0981 - iou\_score: 0.8605 - val\_loss: 0.2813 - val\_iou\_score: 0.6767  
  
  
which iis much faster but actually less precise.  
  
  
maybe the size is too small.  
  
lets try with the original 400 images, but keep the num filters reduced  
  
  
  
loss: 0.0852 - iou\_score: 0.8779 - val\_loss: 0.1993 - val\_iou\_score: 0.7535  
  
  
much better !

IoU= True Positive (TP)​ / True Positive (TP)+False Positive (FP)+False Negative (FN)

lets try to change the unet a bit (play with num of layers and num filters  
  
  
going from 32 to 256 is worse. Dropping the iou by 3 points  
  
  
  
  
Ok I figured out the output images are 608x608  
  
1rst experiement was to train on 400x400, then resize the test image before and after the predicitons.  
  
F1 score : 0695 : Accuracy : 0.848  
  
  
2nd experiment will involve using a patching technique.  
Images to be predicted will be 256  
  
  
  
loss: 0.0756 - iou\_score: 0.8906 - val\_loss: 0.1182 - val\_iou\_score: 0.8451  
  
WOOOOH !  
  
  
  
  
  
We now need data augmentation techniques in order to attain   
  
3rd will be to use best case and try to reduce the unet depth and tweak the drop out rates, and play with the number of filters