BlindNet: Scene Based Knowledge Distillation

Ashutosh Tiwari and **Radheshyam Verma**
Indiana University
Graduate Luddy School Of Informatics, Computing and Engineering
ashutiwa@iu.edu, radverma@iu.edu

Abstract

Object classification is one of the core problem in computer vision. In real world the objects share a relationship with the surrounding. In this project, we attempt to use deep learning techniques to evaluate if a deep learning model can learn the relationship between the object and scene. We attempt to make a deep learning model learn to infer the object from the surrounding. We perform few experiments all of which try to capture this essence some way or the other. Code is made available here.

1 Data

1.1 The Dataset

For this task we will use the coco2017 (Lin et al., 2014) dataset. The coco dataset contains pixel level as well as has the bounding box level details of the objects present in the image. The images vary in size. The training dataset contains 115k images and the validation dataset contains 5k images. There are 91 different classes that are marked in the images. Pixel level class details are also available. But for our purpose we will use the bounding box only.

1.2 Data Preprocessing

We resize all the images to a standard size so that it can be shaped in the form of tensor form when feeding it to the network. We use 0 padding after the resize. Additionally we use RGB shifts, horizontal shifts, Channel shuffle, gray conversion, blurs, color jitters with varying probabilities of their application during the image augmentation. These effects are applied at random with small probabilities. We also rotate the image by small angle with small probability. When processing validation images, we only re-scale the image to a standard size that the deep learning model expects; we do not apply any other transformation.

For experiments 3.3 and 3.4 we use 64×64 gray images. For 3.3 data was processed first and put as a repository here. Because masking images by replacing each pixel with its class id was a computationally intensive task, we first generated all the numpy arrays corresponding to all images and then just loaded them for training. At training time we loaded each image and its target and then did random crop and horizontal flip and resized it before feeding it to model. There were various reasons to choose this small, one of those being training infrastructure at our disposal.

2 Related works

In similar works (Yu et al., 2018) attempts to repaint masked part of the image using gated convolutions. In another works (Suvorov et al., 2021) attempts to remove the masked region indicated by user and inpaints the masked region using fourier convolutions. Our work differs because of the fact that we attempt to predict the class inside the masked region of the image instead of repainting. Thus, We try to learn the relationship between the object and the scene.

3 Methods

3.1 UNET: pixel level prediction

We use UNET (Ronneberger et al., 2015) architecture and make a prediction at each pixel of the bounding box(instead of the object boundary). Similar to UNET the output dimension is 91 x width x length. Additionally we mask one of the object out of several from the image. We supply one additional channel indicating the region where the object was masked. We do this, because when we mask an object, we set the pixels to zero, to differentiate legitimate black pixels such as due to padding or black pixels due to dark regions. So the input to the UNET has 4 channels, 3 RGB channels and one mask channel, as shown in the 1

We use bounding box for masking instead of pixel level segment masking, because we think that

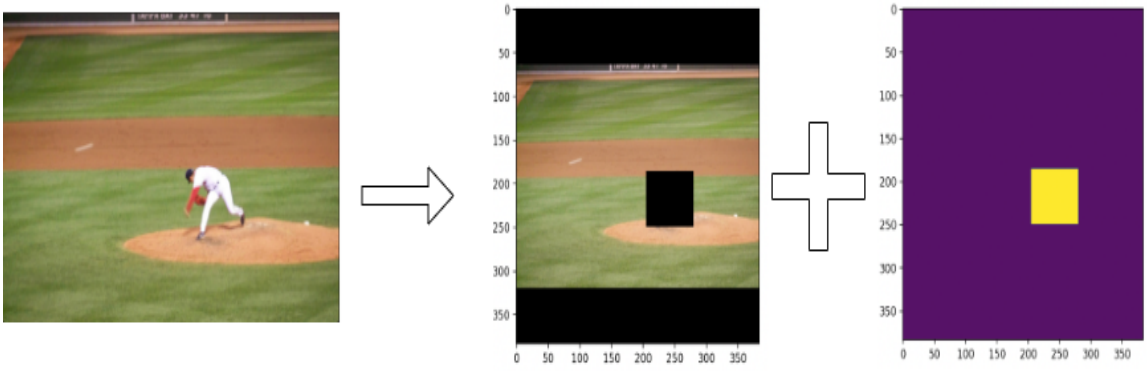

Figure 1: Input image and 4-channel input to the UNET and CNN models

segmenting out the exact pixels gives away the shape of the object that has been masked. Due to this, if multiple bounding box overlap with each other, the network needs to predict both classes. So instead of softmax activation/loss, we use sigmoid for prediction.

The predictions in the network are made for both visible and masked object. Additionally we experiment with weighted loss to prioritize the learning of the masked region. At the inference time, to make a prediction for the whole masked region, we take the sum of all the sigmoid activations for each class. We use this sum as a class prediction in the masked region to do inference.

During inference, we use the trained model and randomly mask the area and let the model infer the class inside the masked region. As during the training, there was an object inside the mask every time, so in this inference we randomly mask the area to see what the model thinks could be present in the region. We change the masked region in the image to see how the predictions change.

The input image size is 384 x 384 to the UNET. In the UNET model, we downsample 4 times and then in the later half of the model we upsample 4 times with corresponding connection from the downsampling CNN layers. We have experimented with two upsampling methods. One is bilinear interpolation, which is parameterless interpolation of intermediate pixels using bilinear upsampling. In the other method we used convtranspose2D layer to increase the dimensionality; which is not parameterless.

3.2 Using simple CNN to predict the masked object

Along with pixel level predictions in previous method, we also used a simpler method where we predict the object present in the mask directly. The output size is C dimensional where C is number of classes. The CNN network that we use here is first half of the UNET network. At the end of the CNN network we add Dense layers. The input is same as that of UNET method. We add additional mask same as that of UNET, to indicate the position of the mask where the object needs to be inferred.

We can see that in this method, the network never finds out what an object looks like without mask. Because all the predictions are with the mask, so even though same object is present in the image elsewhere, the network never finds out that this is the same type of object that is being inferred in the masked region.

3.3 Pixel wise class id prediction

In this experiment we tried to distill knowledge using a Resnet-18 based model. In this case targets were class id masks and input was the image itself, thereby forcing model to predict the pixel wise class distribution over 91 classes, which is the output of softmax over all those. The output from resnet is a 128 dimensional feature map which is then fed into a bunch of MLPs.

Loss function used here is Cross Entropy loss. We did hyperparameter tuning using wandb. We tried two variations of this model. In one variation, there was no weight for any class during calculating loss and then in other case a weight vector was provided to Cross Entropy loss function which are just constants vector which are multiplied to each class component. In both cases after parameters were figured out model was trained using those for 200 epochs. Images were resized to 64×64 on training. Resnet used has pretrained weights and then models are tuned over period of 200 iterations.

This model was different from other previous

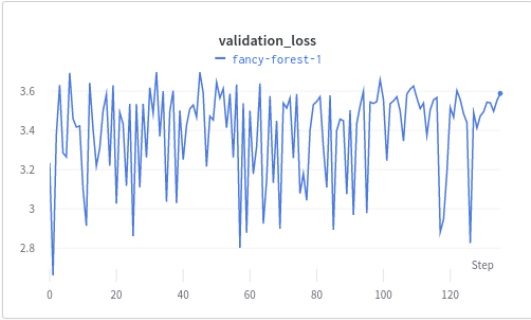

Figure 2: Validation loss for experiment 3.3

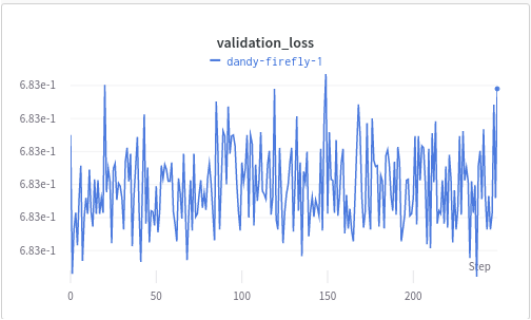

Figure 3: Validation loss for experiment 3.4

models in following sense

- Backbone used was Resnet 18. (He et al., 2015)
- Model is trained on 64×64 gray images.

3.4 Multilabel classification of classes in a scene

In this experiment we forced network to learn all the classes present in a scene. Therefore target values are a hot vector of size 91 which represents all the classes present in a scene. Loss used here was Binary CrossEntropy loss. This also had two variants very similar to previous case. One had class weights. Similar to previous experiment, pretrained Resnet-18 is used which is then fine tuned.

4 Results

On running UNET and simple CNN with multiple methodologies, we found that the inference produced by our network is as good as a random prediction. To assess the goodness of a prediction, we checked the top-1, top-2 ... top-C scores. We plot these top scores against the C as shown in figure 4. In this case we know that if there are C classes then top-C score will be 100%. So a random guess will be a straight line. However we also noticed

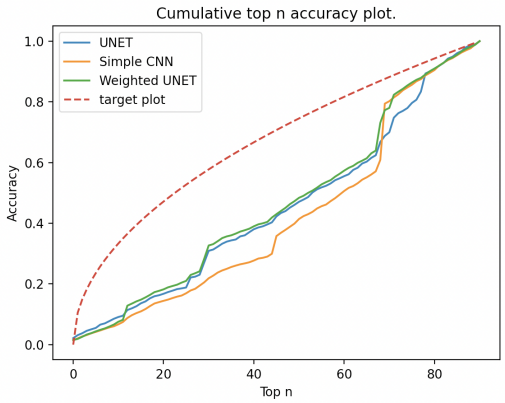

Figure 4: Top-n vs accuracy plot. The ideal plot indicates the accuracy plot we expect to see when the model infers majority of the objects in the early top-n classes.

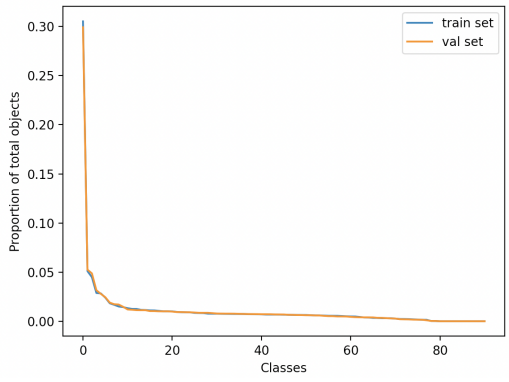

Figure 5: Class proportion in dataset in decreasing order indicating the person class(1st place) taking up significant proportion of overall objects.

that there is significant class imbalance. Because most photos are taken by humans. The images predominantly contain people or crowd. Due to which the person class is significantly more represented in the dataset. The figure shows the distribution of classes in validation and training dataset in the decreasing order of their occurence.

We found that both upsampling methods, convtransepose2D and bilinear interpolation perform similarly. The same results continue for UNET network with weighted loss. Where we weighted the loss at the masked region more compared to unmasked region. Some of the issues with the network that we think could be that some of the masks are quite large and cover most of the image, while some masks cover other objects too. In this case masking one object masks the other object as well. We think this could be interfering with the training. In the future work we intend to resolve these issues by limiting the mask to be under a certain ratio of the image.

In some way our results are better than majority prediction, because in a majority prediction, 'person' class, which is present more than 30% of the time, will predict 'person' class every time. While our model does not predict person class and instead predicts other classes equally proportionally and the top-n accuracy plot is a straighter line instead of looking like class distribution plot.5

For 3.3 and 3.4, validation loss is shown above. Results were okay. There were few images with small number of classes for which results are good. But in cases where number of classes are very big, because we resize I think we loose some information mostly of classes which are small in size.

5 Future Work

From here we have multiple possible experiments that we would like to perform to make the model work better. We want to expand into class level analysis on how good each individual class performs. With respect to training, we want to try out partial masking first and limiting the mask to a certain ratio of the image. From partial masking, during the training we may increase how much portion we mask during the training and finally perform ratio level analysis on masked region. Additionally we want to borrow techniques from language processing like transformer to make the masked region classification, similar to how masked tokens are predicted in the NLP.

For last two experiments probably we could have tried all channels with full sized images and then use very low learning rate for a very long time.

6 Conclusion

The masked class prediction is very challenging task. We wanted to apply techniques similar to NLP where the model predicts the missing tokens. However in the visual world the most of the scenes do not follow a grammar like the languages. However, we believe we can still apply a lot of techniques from the language processing into the visual processing and achieve close to ideal curve.

References

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597.

Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. 2021. Resolution-robust large mask inpainting with fourier convolutions. *CoRR*, abs/2109.07161.

Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. 2018. Free-form image inpainting with gated convolution. *CoRR*, abs/1806.03589.