Dataset Description

The dataset for this deep learning assignment consists of images of four different animals: cats, dogs, elephants, and tigers. Each image in the dataset has dimensions of 256x256 pixels thus no resizing was necessary as such during preprocesssing. Additionally, each image contains two masked areas, which are represented as 75x75 boxes. The goal of this assignment is to predict the output for these two masked areas in each image.

The dataset is organized into several sub-folders, each corresponding to one of the four animal categories. Within each sub-folder, there are multiple image files in the JPEG/JPG/PNG format, each with a unique filename. The filenames follow a consistent naming convention that allows for easy identification of the animal type and the image number. The details of the bounding boxes for each image are given in a csv file in the folder that contains masked images.

Data Processing

To improve the robustness of the deep learning model on this dataset, several data augmentation techniques were applied. These techniques include random cropping, random jitter, and random flipping of the images.

Random cropping involves selecting a random portion of the image and resizing it to the desired input size. The images were resized to 286x286 and then cropped back to 256x256, hence producing the desired cropping. Random jitter involves adding small random variations to the pixel values of the image. This technique can help the model learn to be less sensitive to small changes in lighting or color. Random flipping involves horizontally flipping the image randomly with a 50% probability. This technique can help the model learn to recognize objects regardless of their orientation in the image. By applying these data augmentation techniques, the input size was inflated from 7000 to 15000 before starting training on it.

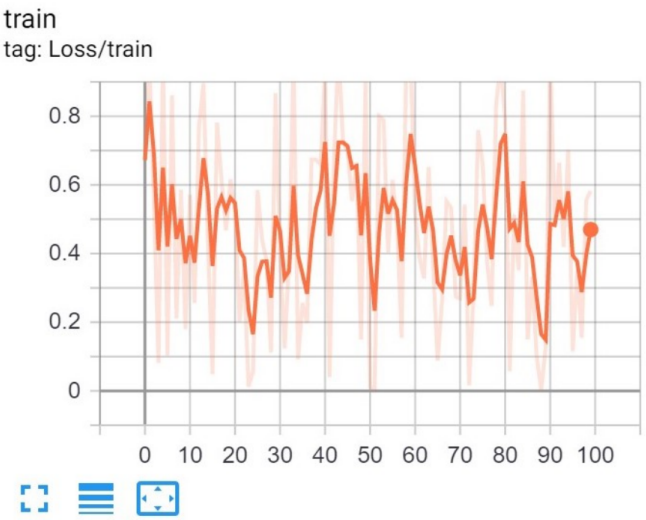
Model and Architecture

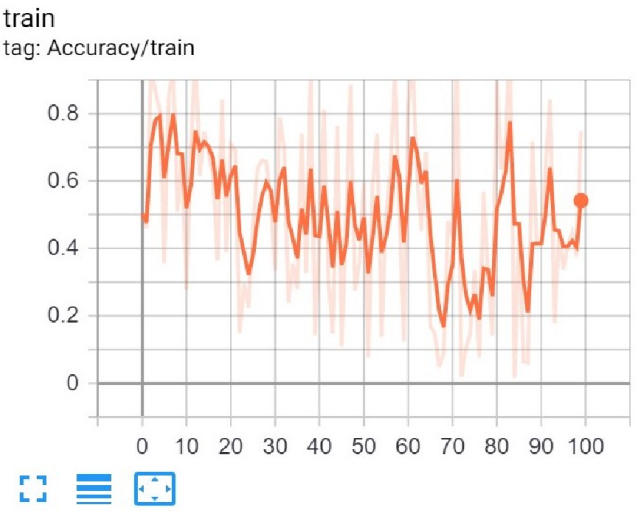
We have used pix2pix model, which is a state-of-the-art conditional generative adversarial network (GAN) model that is capable of learning a mapping between input and output images. In the case of this particular dataset, the input images contain two masked areas, and the corresponding output images contain those same masked areas but with the predicted animal type.

The Pix2pix model is comprised of two main components: a generator network and a discriminator network. The generator network is an encoder-decoder architecture with skip connections. The encoder network compresses the input image into a lower-dimensional representation, while the decoder network generates the output image by upscaling this representation. The discriminator network is a CNN that takes in pairs of images, either a real input/output pair or a generated input/output pair, and determines whether the pair is real or fake. The discriminator is trained to

correctly classify the input/output pairs, while the generator is trained to produce output images that fool the discriminator into thinking they are real.

Loss Plots and Interpretation





In a GAN, the generator loss measures how well the generator network is able to produce high-quality output images that are indistinguishable from real ones, while the discriminator loss measures how well the discriminator network is able to distinguish between real and generated input/output pairs. The balance between these losses is important for successful GAN training. In the picture, the generator loss on the left and discriminator loss on the right show the progression of the training process towards convergence, where the generator produces high-quality output images that fool the discriminator.
Both the losses should decrease together. One of our previously trained models failed as the discriminator won, which is characterised by the loss decreasing to very low levels. At this point the only solution is to retrain the GAN. The plots from the failed training are given below.

Score and Scope of improvement

Currently we have a score of 0.25390 on Kaggle, and further ideas for improvement include trying out stable diffusion models and using a Mask Aware Transformer, which is a state of the art transformer model to get better predictions and actually use the bounding boxes to produce output in a specified region instead of mapping the complete input to output.