Technical Report for Summer Internship 2019

Biomedical Imaging Algorithms, Czech Technical University in Prague

Vatsal Goel
Indian Institute of Technology, Guwahati
vatsal29@iitg.ac.in
vatsalgoel.98@gmail.com

About

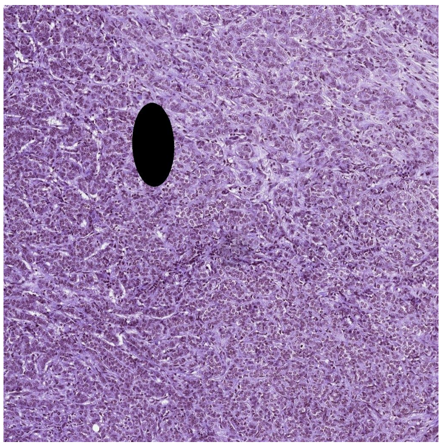
This report summarizes the work done during my summer internship, 2019 at Biomedical Imaging Algorithms, Czech Technical University in Prague under the guidance of Jan Kybic and Jan Hering. First we started at implementing Aicha Bentaieb's code for predicting cancer with a recurrent visual attention model for histopathology images, link - [http://www.sfu.ca/~abentaie/papers/miccai18.pdf]. The source code given by the authors was incorrect and incomplete. We started with some small fixes to the code. The major issue was that the location for successive glimpses was not updating during training. So we tried to solve that and were finally able to do that by changing the gaussian glimpse extraction code. During this debugging we came across some important results which have been listed. Further, we found that Aicha's model represents the EDRAM architecture closely, link - [https://arxiv.org/abs/1706.03581], so I modified Aicha's code as the EDRAM architecture and tested our data on it.

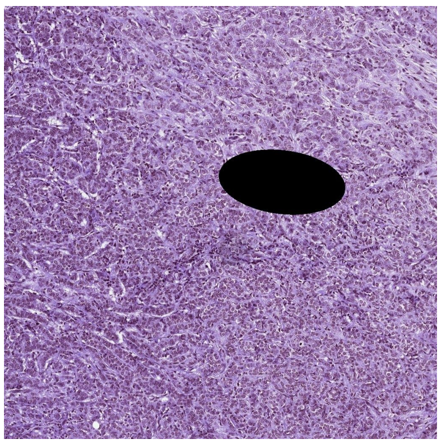
Unfortunately we were unable to replicate Aicha's results mentioned in their paper. We also wrote them by email but had no reply. So we switched to a different approach, using multi levels U-Net architecture with reinforcement learning. I tested the single level U-Net architecture for level 1 and 2 while Jan worked on combining the different levels. Next, I worked on testing a pattern dataset (consisting of stripe patterns for negative image and checkerboard pattern for tumor image) created by Jan on the EDRAM model for a baseline comparison to our multi level U-Net architecture. I also created a new pattern dataset which is more difficult to train (multiple stripe patterns for negative image and multiple stripes and checkerboard pattern for tumor image). Lastly, I converted these new pattern images to 3 channel images so as to be able to use the pre-trained Inception Network for training.

All the code for my work has been pushed to GITLAB under the Biomedical Imaging Algorithms group, link - [https://gitlab.fel.cvut.cz/biomedical-imaging-algorithms] . I have provided the information about the repository and the branch for every piece of code. I have included some illustrations of the results in this document and in depth analysis of each model can be found in tensorboard logs. I have provided the path for every tensorboard log in this report.

Small fixes at Aicha's code

- Fixed initial stride from 0.5 to 1 to get the whole image - in declaring gt inside function init_loop in the file recurrent_attention_utils.py
- Changed labels from 1/-1 to 0/1 in the training set (stopped accuracy from decreasing over time)
- The tf.resize function is broken, wrote own script (image_resize.py) to resize using OpenCV and stored them in qqgoel/deb_testing - not used in the final script
- Tweaked the pre-processing for train and test, used the resize_img_keep_aspect_ratio function declared in image_ops.py
- /local/temporary/qqgoel/attn/logs/inception_recurrent_attn_debug_7 - tensorboard log, getting decent results for came16_debug_train.txt dataset
- Created own synthetic dataset having black ellipses at a random place with a random orientation angle, for easier training. 130 normal and 130 tumor images of size 1024x1024 each. Stored in /qqgoel/synthetic_dataset
- Trained 2 models on synthetic dataset with 0 and 3 glimpses respectively. Achieved training accuracy of 0.994 and 0.996 . Tensorboard log for 3 glimpse model in /local/temporary/qqgoel/attn/logs/inception_recurrent_attn_debug_latest. All the losses approach zero except for location loss. The location loss settles on 6 but it does not make a difference to the model because gamma decays by 0.01 after every 1000 steps making the contribution of the location loss very little.
- Trained another model keeping gamma 1 with 3 glimpses. No change to the location loss, still remains at 6 for synthetic dataset.
- Also tried by using tanh (to get range from -1 to 1) as activation function for location parameters instead of their sigmoid activation, again made no difference for synthetic dataset.
- Made another synthetic dataset with varying ellipse sizes this time for tumor. This





  dataset is stored in /datagrid/Medical/microscopy/attn_testing/synthetic_data. Achieved training accuracy of 0.99 for 3 glimpses. Tensorboard log in zorn in /local/temporary/qqgoel/attn/logs/inception_recurrent_attn_debug_syn2.
- **Imp Point -** Since the final prediction is computed using the feature outputs of all the glimpse feature outputs x[1:p], the model doesn't necessarily have to zoom in on the tumor region in successive glimpses. If the input image (glimpse 0) is good enough to classify, it can just assign the weights of the rest of the glimpse features to zero or don't care.
- Increased glimpse size to 299, the default size for inception networks

Location update debugging

On printing the new location after every iteration, the center is constant at (-0.5, -0.5), and after the last fully connected layer, is computed as -

$$new\_loc = lower\_bound * old\_loc + upper\_bound$$

where *lower_bound* = -1 and *upper_bound* = 0.5. This means that *old_loc* = 1, hence the <u>output of the last layer is very large</u> for sigmoid to output 1.

Location Loss is 6 because since the location does not update, its e^0 = 1 for x and y both, so total = 2. It sums for all glimpses so 3 glimpses give loss = 6. Also, location loss is not computed between L_0 and L_1.

<u>The hadamard product of glimpse features and location features is almost zero - </u>On observing the distribution of the **what_and_where** combination after the init_loop over various models, it stays pretty much close to zero. This could also be the reason behind no updates in locations over time.

<u>The 3 glimpse model is same as 0 glimpse model - </u>Trained 2 models on the equally distributed main dataset, one with 0 glimpses and 1 with 3 glimpses. Changed lr to decay after 5000 steps and gamma to remain at 1 for the 3 glimpse model. Both the models produced almost identical results after 10000 steps. This is also justifiable by the fact that the location for the glimpse still does not update so the 3 glimpse model has trained itself to classify from glimpse 0 only.

<u>Location updates on using Gaussian Glimpse Extraction</u>

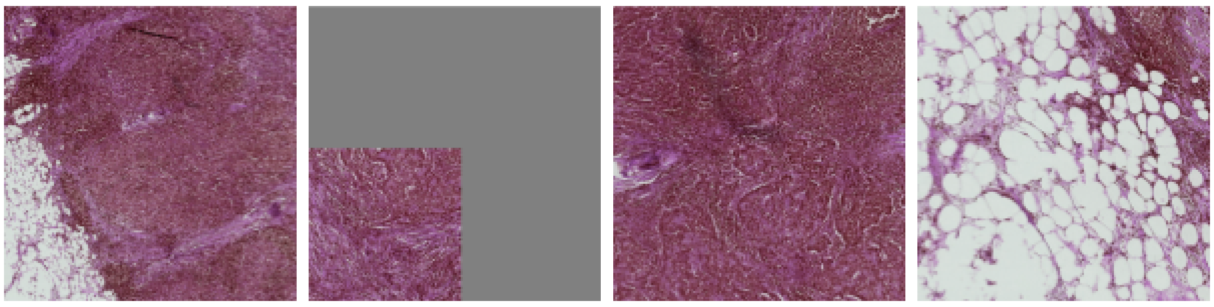
Possible ideas to try / Things to keep in mind

- Tensorflow's conv and FC layers have activation relu and initializer xavier by default.
- Tensorflow's softmax_cross_entropy loss applies softmax and then computes loss.
- Do NOT use tensorflow's inception pre_process function.
- **WSI Prediction Idea -** Instead of aggregating all patches during testing time, we can get the results for all individual patches and then just use np.argmax() so that even if 1 tile has a tumor its identified as a tumor image.
- Location update based on GT values of tumor tissue
- Could use IoU loss for supervised location training. The drawback of the conventional losses in bounding box regression is that the bounding box transformation parameters (tx, ty, tw, th) are optimized independently although they are in fact highly intercorrelated.

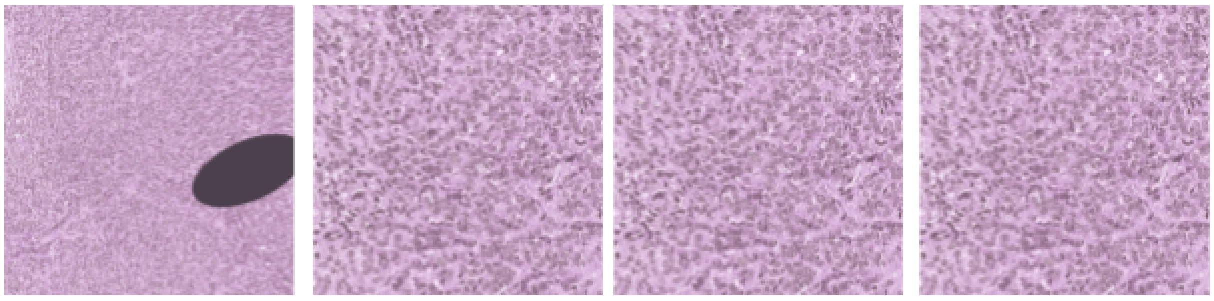
Important Experiments and Observations

Pre-trained Inception Network fails for skewed data - Trained 2 models, both having 0 glimpses i.e. only looking at the entire image once, but one having balanced data and one with skewed data. With balanced data achieved good results with training accuracy of 0.81, PREC = 0.6, REC = 0.73 and F1 = 0.65. The skewed data model did not work. Tensorboard logs in zorn in /local/temporary/qqgoel/attn/logs/inception_recurrent_attn_main_half_0 and /local/temporary/qqgoel/attn/logs/inception_recurrent_attn_training_2.

The 3 glimpse model works for more balanced data - I created another dataset, from the main dataset, extracting patches with min coverage of 0.5 and taking balanced number of tumor and normal tiles. However, I took way more images in this dataset. Previous balanced dataset had a total of 223 training images, this dataset has 1505 training images. The dataset is stored in /datagrid/Medical/microscopy/attn_testing/bal_data/training. After the glimpse size was increased to 299, the model with fewer images gave 0 f1 score but the other model (lr decays at 7000 and gamma at 5000) works giving f1 score of 0.66. Precision is at 0.8 and recall at 0.57. Training accuracy is 0.96. These results do come close to the paper results, will try more by tweaking hyperparameters. The rank loss used for this model La is not as per their original code, I modified it to be according to the given equation 4 in the paper. Tensorboard log for working model in /datagrid/temporary/qqgoel/jhtest_bal3_N_299


Original Image	Glimpse 1	Glimpse 2	Glimpse 3

Mean Square Error loss using GT center values for location does not work - On the synthetic dataset with varying size ellipses, on passing the gt center values of the ellipses for tumor images and using mean square error loss which is computed as the l2 norm or the euclidean distance between the predicted center location and the ground truth center location, the network still does not zoom in onto the tumor area. Also, the location is not changing for the 3 different glimpses.


Original Image	Glimpse 1	Glimpse 2	Glimpse 3

New area location loss using tumor masks works - Created new dataset by extracting patches with min coverage of 0.5 and their tumor masks if tumor images. Currently using only 600 images. For the normal tiles, the mask is an image with all zeros. Using location loss given by

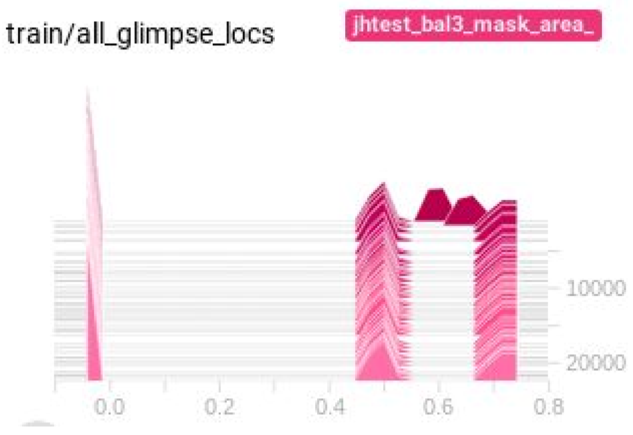
$$location\ loss\ =\frac{1}{N}\sum_{i=1}^{N}-label^{(i)}*ln\left(\frac{\sum_{r=1}^{R}\sum_{c=1}^{C}(mask\ glimpse)_{rc}^{(i)}}{glimpse\ height*glimpse\ width}\right)$$

Note: All the mask pixel values should belong to {0,1} for the above equation to work

Where *mask glimpse* is the gaussian extracted glimpse of the tumor mask similar to the extracted glimpse of the input image.

The 3 glimpse model is training, but the locations are $L_1 = L_2 = L_3$. There seems to be some error with the LSTM cell as the hidden state does not seem to update giving the same location. After training for 23000 epochs the training accuracy is 0.95, precision is 0.9, recall is 0.62 and f1 score is 0.73.

Tensorboard log in /datagrid/temporary/qqgoel/jhtest_bal3_mask_area_



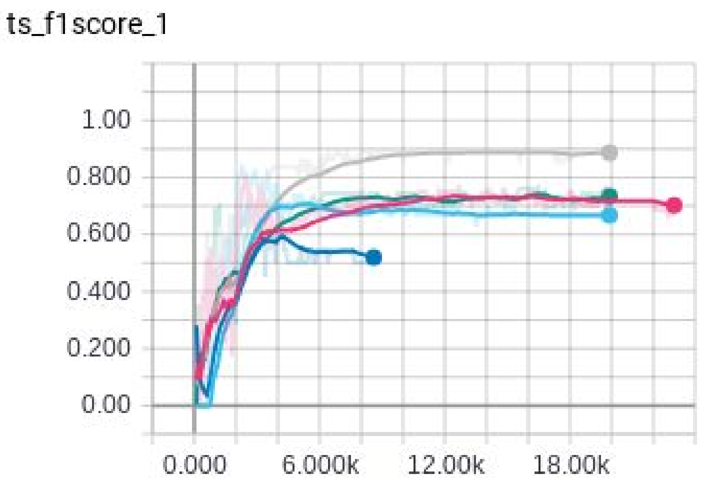
CODE - The code can be found in the repository Visual Attraction under the Biomedical Imaging Algorithm group in the **branch vg-testing.** Link - [https://gitlab.fel.cvut.cz/biomedical-imaging-algorithms/visual-attraction/tree/vg-testing]

Using combination of both location losses - Using location loss given by

$$location\ loss\ =\ \alpha*L_\Omega+(1-\alpha)*L_d$$

Where, L_Ω is the area loss defined above and L_d is their distance loss defined in the paper. I conducted 5 experiments with the values of alpha being {1, 0.7, 0.5, 0.3, 0}. The dataset is same as previous one containing 600 images. The model with **alpha = 0.5 works best**. Tensorboard logs in the same order are at:

1. /datagrid/temporary/qqgoel/jhtest_bal3_mask_area_ (magenta)
2. /datagrid/temporary/qqgoel/jhtest_bal3_mask_area_0.7 (green)
3. /datagrid/temporary/qqgoel/jhtest_bal3_mask_area_0.5 (grey)
4. /datagrid/temporary/qqgoel/jhtest_bal3_mask_area_0.3 (blue)
5. /datagrid/temporary/qqgoel/jhtest_bal3_mask_area_0 (light blue)



Training main data using bal3_0.5 weights - Trained the 3 glimpse model on the main dataset by loading the weights from the trained model on the bal3 dataset for alpha = 0.5 which had produced the best results. Hyperparameters are - Initial lr = 1e-5, decaying at every 10k steps by 0.1, gamma = 1, decay same as lr. Couldn't get the same results as on the bal3 dataset.

Attention models converge randomly - I trained the alpha=0.5 model on the bal3 dataset again many times, but was **unable to replicate the results** achieved the first time. It seems the model converges properly very rarely and otherwise settles on some local minima.

Using EDRAM architecture - I changed Aicha's code to implement the EDRAM architecture, link - [https://arxiv.org/abs/1706.03581]. For the context network, I took the glimpse 0 feature vector given by the inception network, passed it through a Fully Connected layer and used the output vector as the initial state for the LSTM cell for the glimpses. I used the Gaussian glimpse extraction, upon using the spatial transformer extraction the location again settles on extremes and does not learn at all. For this model the losses used are - glimpse level loss, aggregated glimpses loss and using the previous combination for location loss with alpha = 0.5.

Tensorboard Log in /datagrid/temporary/qqgoel/jhtest_bal3_mask_area_edram_context
The Results are not close to Aicha's results with F1 score = 0.57.

	Train Accuracy	Precision	Recall	F1 score
Aicha's model	0.97	0.98	0.82	0.95
EDRAM	0.85	0.5	0.67	0.57

CODE - The code can be found in the repository Visual Attraction under the Biomedical Imaging Algorithm group in the **branch EDRAM.** Link - [https://gitlab.fel.cvut.cz/biomedical-imaging-algorithms/visual-attraction/tree/EDRAM]

U-Net Architecture

U-Net is a more elegant architecture, stemming from the so-called "fully convolutional network". The main idea is to supplement a usual contracting network by successive layers, where pooling operations are replaced by upsampling operators. Hence these layers increase the resolution of the output. What's more, a successive convolutional layer can then learn to assemble a precise output based on this information. More Information - https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

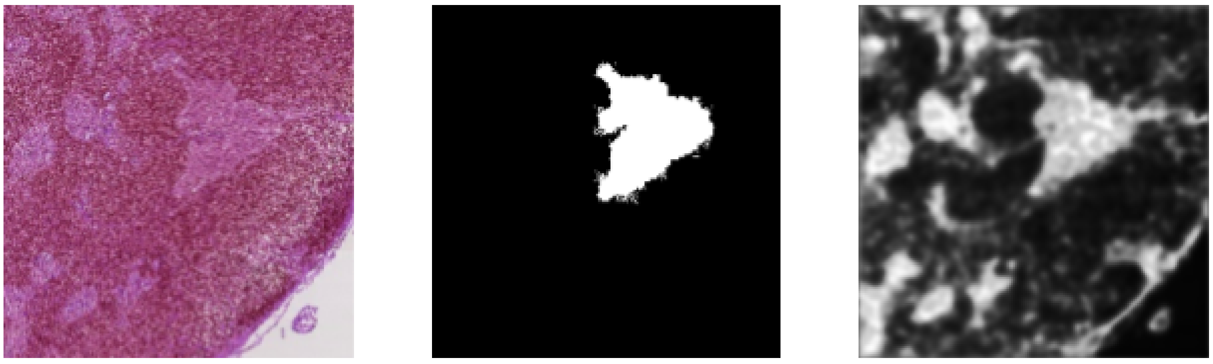
Since our dataset consists of very high resolution images, we think looking at different levels of magnification should produce better results than just looking at one level. Also, if the tumor region is very small in an image compared to healthy region, successively zooming in on the tumor region should make it easier for the classifier to judge.

Single level U-Net architecture

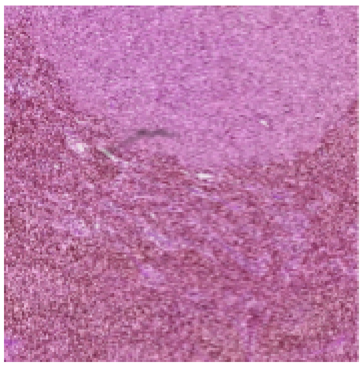
CODE - The code can be found in the repository Spatial Attention under the Biomedical Imaging Algorithm group in the **branch single-level-unet.** Link - https://gitlab.fel.cvut.cz/biomedical-imaging-algorithms/spatial-attention/tree/single-level-unet

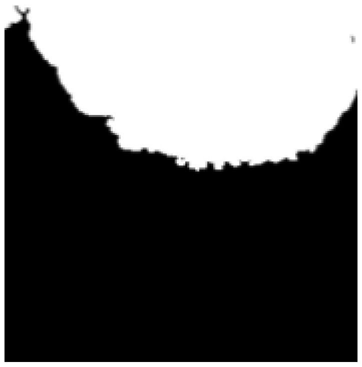
Found a working implementation of the U-Net architecture, link - [https://arxiv.org/abs/1505.04597]. I ran the single level unet architecture on the main dataset. I used only the tumor tiles and their respective tumor masks, which are resized to (512, 512). The dataset is stored in /datagrid/Medical/microscopy/attn_testing/unet_testing. I trained the model for extracting images at both level 1 (the entire image downsampled to (256,256)) and level 2 (random cropping of (256, 256) and resizing) from the original (512, 512) image. The tensorboard logs respectively are:

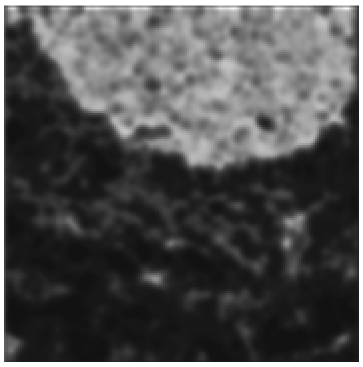
1. `/datagrid/temporary/qqgoel/vat_testing/train_unet_lev1_bal_lr150`



2. `/datagrid/temporary/qqgoel/vat_testing/train_unet_lev2_bal_lr150`


Original Image


Original Mask


Output Image

For this model, if the learning rate is kept high for too long (decay at ~400 steps), the model diverges. After a few experiments, I found that the model works best for learning rate decaying at every 150 steps by 0.1. The checkpoints are at:

1. /datagrid/temporary/qqgoel/vat_testing/checkpoints/checkpoint_lev1_unet_bal_lr150.ckpt-50
2. /datagrid/temporary/qqgoel/vat_testing/checkpoints/checkpoint_lev2_unet_bal_lr150.ckpt-25

Further, I added support for restoring session from a checkpoint and for evaluating on validation data.

Multi-level U-Net architecture

Found a solution for sampling the centers after every iteration from tensors instead of numpy arrays, so that we can have successive levels of the architecture in the same graph.

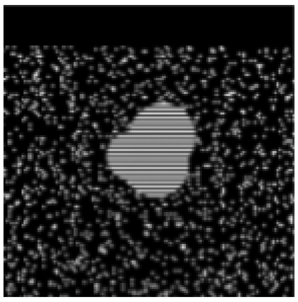
Pattern Data

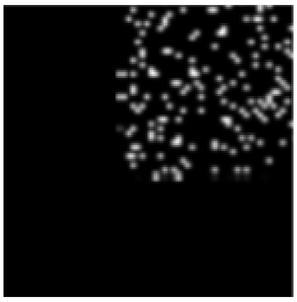
CODE - The code can be found in the repository Spatial Attention under the Biomedical Imaging Algorithm group in the **branch edram-pattern-testing.** Link - https://gitlab.fel.cvut.cz/biomedical-imaging-algorithms/spatial-attention/tree/edram-pattern-testing

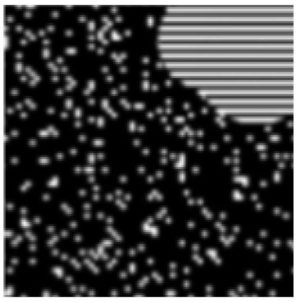
Testing pattern data on EDRAM

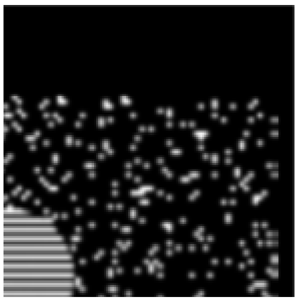
I trained the EDRAM model on the pattern data created by Jan described in the text file /datagrid/Medical/microscopy/vat_testing/pattern_bwdata_fieldsize_4_training.txt. The glimpse used for this model was 128 and there were 3 glimpses. Lr was set to decay by 0.1 at every 4000 steps and gamma by 0.1 at every 6000 steps. However, both the models use gaussian glimpse extraction instead of the spatial transformers. I trained 2 models with differences in data loading.

1. This model crops the original 1024 x 1024 image to a 512 x 512 image around the pattern region. The results on this model were good with F1 score = 1. The tensorboard logs are in - /datagrid/temporary/qqgoel/syn3_level1_edram

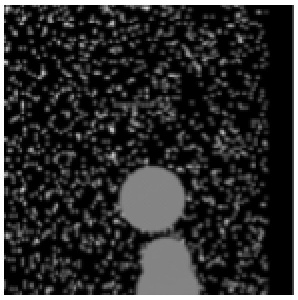

Original Image


Glimpse 1

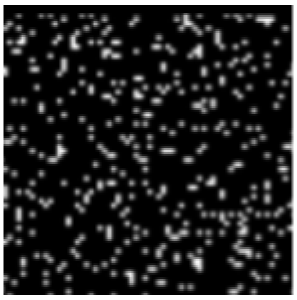

Glimpse 2

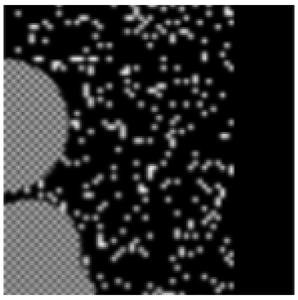

Glimpse 3

2. This model crops the original 1024 x 1024 image to a 512 x 512 image around the pattern region but it also has an offset on the Ground Truth center in the range [-0.2, 0.2). The results on this model were not as good as the previous model with F1 score = 0.92. The tensorboard logs are in - /datagrid/temporary/qqgoel/syn3_level1_edram_offset

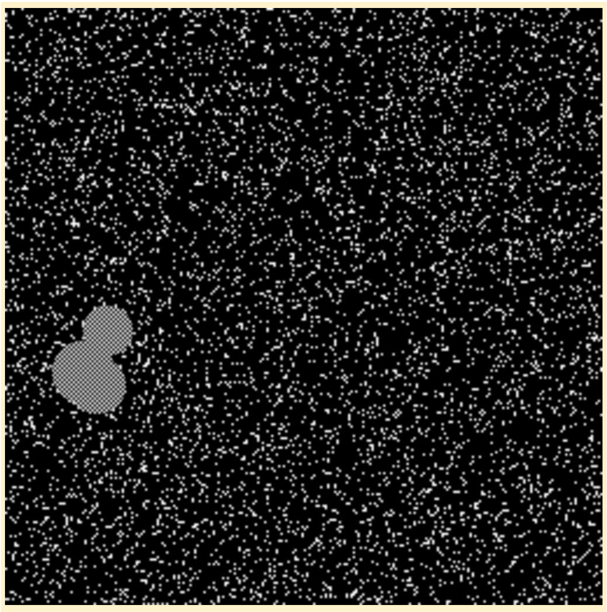

Original Image

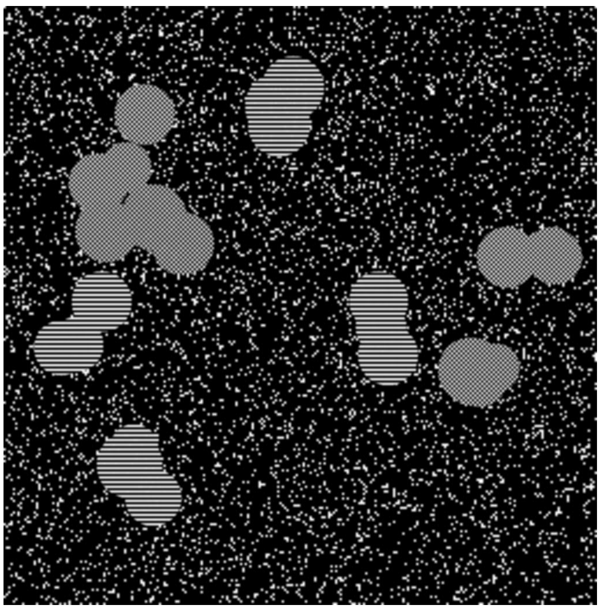
Glimpse 1


Glimpse 2


Glimpse 3

Next, I created another pattern dataset which is difficult to train. Previously, the dataset had an image containing only 1 set of stripe pattern or 1 set of checkerboard pattern. In this dataset a normal image has multiple sets of stripes pattern and a tumor image has multiple sets of both stripes and checkerboard patterns. This dataset also has the added advantage that if we random crop regions from the image we should get at least one pattern in each cropped region instead of just noisy background. The dataset is given by the text file - /datagrid/Medical/microscopy/attn_testing/patterns_2/pattern_bwdata_fieldsize_4_training.txt

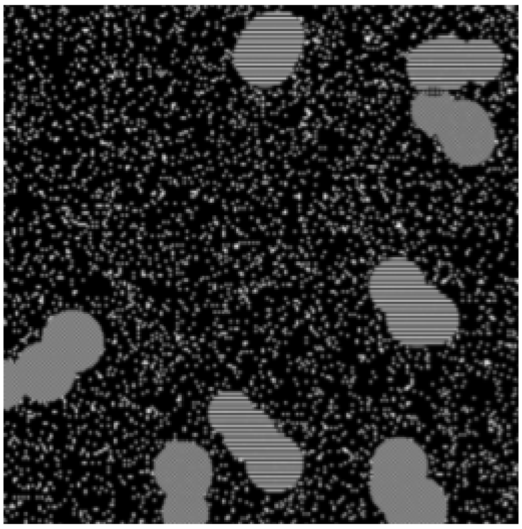

Previous Tumor Image

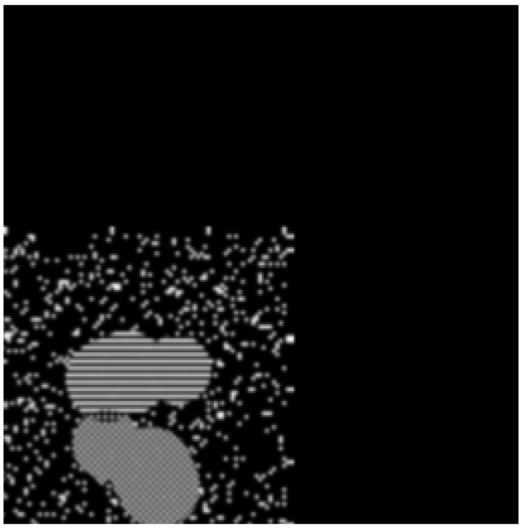

New Tumor Image

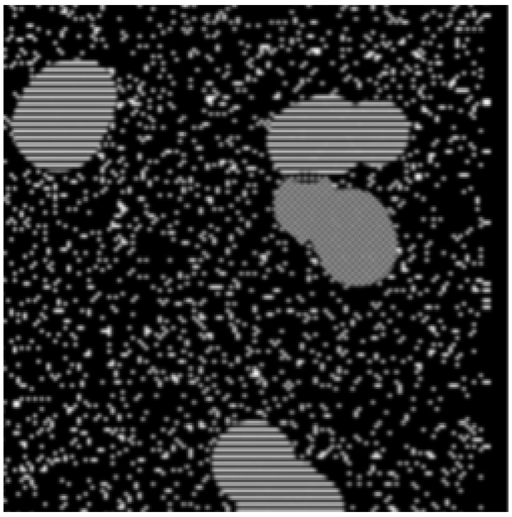
Testing pattern 2 data on EDRAM

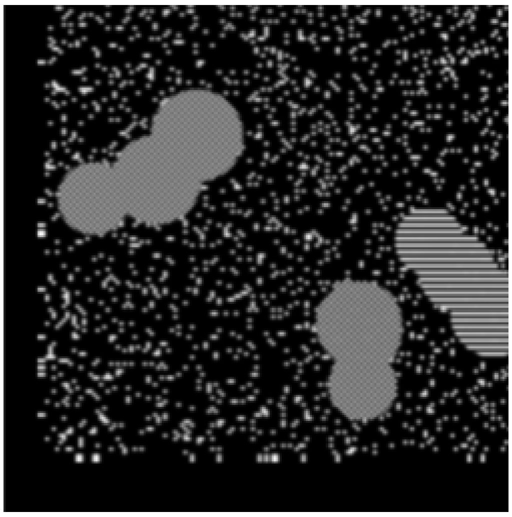
I trained 2 EDRAM models on the new pattern dataset shown in the above image. The glimpse used for this model was 299 and there were 3 glimpses. Lr was set to decay by 0.1 at every 4000 steps and gamma by 0.1 at every 6000 steps. However, both the models use gaussian glimpse extraction instead of the spatial transformers.

1. There is no cropping in this model, the original image is just downsized. The results are not as good as previous dataset with F1 score = 0.65. The tensorboard logs are in /datagrid/temporary/qqgoel/syn3_level1_edram_pattern2_

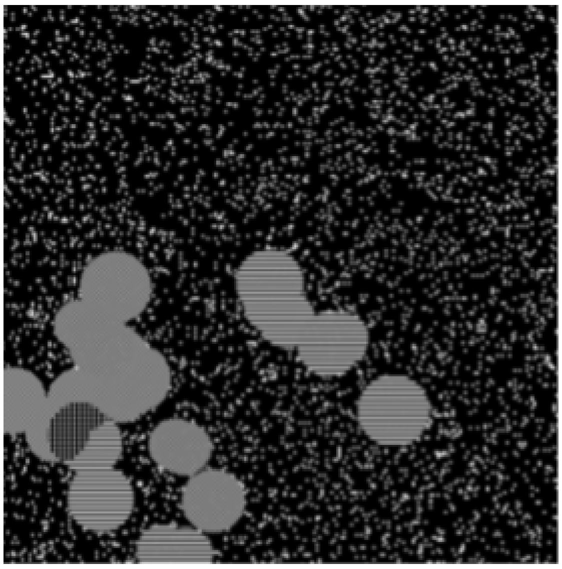

Original Image

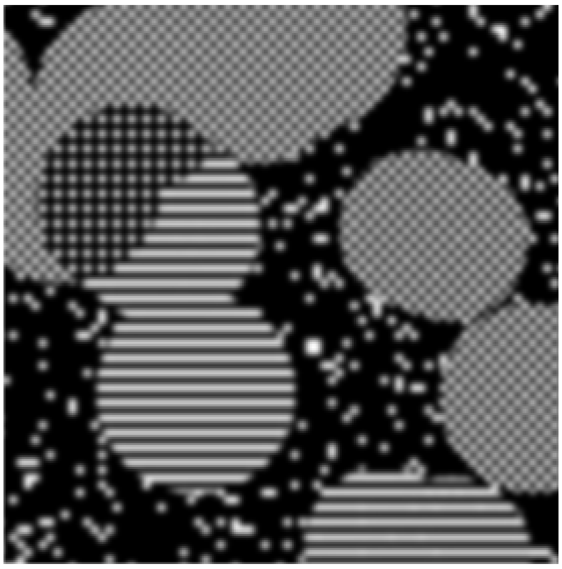

Glimpse 1


Glimpse 2


Glimpse 3

2. For this model, I changed the input images to 3 channel images. The images are still black and white but they have 3 channels. This change enabled me to use pre trained Inception Network. This model converged very well reaching a training accuracy of 1 very quickly and had an F1 score = 0.99. The tensorboard logs are in /datagrid/temporary/qqgoel/syn3_level1_edram_pattern2_rgb


Original Image


Glimpse 2

Note - Glimpse 1 and 3 don't hold any important information for this image hence I didn't attach them.

Result Comparison on old and new pattern data

	Training Accuracy	Testing F1 score
Pattern 1	0.934	1.0
Pattern 1 with offset	0.986	0.92
Pattern 2	0.961	0.69
Pattern 2 with rgb	1.0	0.97