ISIB dataset (2017 has 2000 images, 2016 has 900 images)

**specifics about the dataset:**

1. high intra-class variance: the images belonging to either of the class(cancer/non-cancer) has many types of images. So if given a image, it cannot be generalised that all images of that class will look like that.

2. low inter-class variance: the images belonging to the two classes look very similar. (example cats and dogs are easily distinguishable but whether the lesion is malignant or benign is very difficult to tell.

3. skewed proportion of images in classes: 80% of the images are benign and 20% are malignant.

**how is this dataset similar to Imagenet dataset:**

1. both are clicked in natural light, therefore the concept of edges and corners are still intact.

2. both are labeled.

3. Intra-class variance is high in both.

**how is this dataset different from Imagenet dataset:**

1. Number of images: imagenet has 1.5million images while our dataset has 900/2000 images.

2. Imagenet has high  inter class variance.

**how doctors classify cancer**

1. ABCDE rule: Asymmetry, Border irregularity, colour non uniform, diameter greater than 6mm, evolution in color, size

2. i don't remember.

**Most methods used earlier in image processing:**

1. Either collected features wrt ABCD rule or usual feature detection methods like Bag-of-visual-words, sparse coding, (read goodReview paper for more.)

**Problem with these methods?**

1. Could not model ABCD rule well.

2. High Intra class variance, difficult to find problem specific features because different types of images in the same class have different results on applying filters or methods.

**Good things about these methods?**

1. even though inter-class variance is low, it was not an issue because classifiers like SVM(support vector machines) are designed specifically to handle this type of problem.

2. low compute time and minimum hyper parameters therefore, the researcher had the idea about what is learnt by the model and on what basis does it classify image as cancer or not.

**Why deep learning based classifier?**

1. our dataset similar to imagenet. Therefore features learn't on it can be used on the cancer dataset (transfer learning and fine tuning of pre-trained model.)

2. our dataset is similar to imagenet so the algorithm that works well for imagenet, a mini version of those can be useful on our dataset.

**metrics for evaluation?**

1. Accuracy: depends on the proportion of images, therefore never directly accepted as the only metric.

2. Sensitivity

3. Specificity

4. RUC AUC : Area under the curve

5. Jaccard Coefficient

**Which papers did we implement?**

1.Transfer learning using caffenet: 76% accuracy best so far

2.Transfer learning using resnet: don't remember

3.Sparse Coding: didn't give output for all images because the method involves finding the minima of an objective function. For some images minima could not be found so it gave NaN as output..so not a reliable method. Results are in ppt.

4. Trained a resnet-50

**After this, we did**

1. Visualization: We ran the [<http://yosinski.com/deepvis>] model which uses deconvolution and unpooling layers to extra info from a trained model by undoing what the convolution layers have done and then see the effect of each layer of the model on the input image. We found that

a. initial layers of the trained model learn't to extract either background or foreground,

**Some thumb rules in machine learning,**

1. Best results are obtained when the model is as complex as the dataset.
2. If model is less complex than the dataset, then underfit occurs.
3. If model is more complex than the dataset and poorly regularised, then overfitting occurs.

Therefore, the best bet while dealing with small size dataset is that let the model be complex than the dataset but it is regularised well so that it gives best results.

**How to find complexity of model?**

It is subjective. In some cases it is objective, for example, if the datapoints are in one dimension and along a line then, using a order two equation then it is said that model is more complex than the dataset.

In deep learning (or neural network) context, the complexity is defined by the number of trainable parameters.

**Merits of Residual Networks:**

1. Vanishing Gradient Problem solved: As the model becomes deeper, it extracts more high level information but it was observed that after a point it was not able to improve performance. The reason suspected is that the gradients passed to the deepest layers via backpropogation’s chain rule become negligible when the layers are very deep. So feeding the output of previously computed layer via identity mapping solves this problem.
2. Exploding gradient problem solved: (Identity mapping in residual networks paper explains) If the identity mapping is scaled before adding then, if the scale is <1 then it causes vanishing gradient problem and if scale >1 then it causes exploding gradient problem, ie, a small change in the value of weight might cause huge change in the performance, and hence the model doesn’t stabalizes. Therefore, identity mapping with scaling =1 is best suited as in resnet.
3. Internal covariate shift problem solved by Batch normalization (not by residual but worth mentioning): Input image is normalized based on the mean of complete dataset and not the batch which nullifies the effect of normalization in the first place because generally datasets are multi-modal (many images are of one type and there are multiple types. In other words, the probability distribution of images has many maxima over the images of similar type) Batch norm finds local mean wrt to the batch and hence, stabalizes the whole network while training as well as testing.
4. Degradation problem: Deeper model performed poorly than shallow. (reason: 0 mapping is easier to learn than 1 mapping)

**Some design aspects we considered before deciding on butterfly model-**

1. Model has to be deep as deeper models are expected to extract high level info which is needed in our case because inter-class variance is low so any low level based discrimination is going to perform poorly.
2. The number of trainable parameters shouldn’t be too much because we have limited gpu as well as it leads to overfitting.
3. The expected reason why residual networks approximate a function well is because the residual part just learns to slightly change the input signal (identity mapping) so that it matches the function we are trying to approximate. In one paper it has shown that it is easier to learn 0 mapping than identity mapping) so the residual part fine tunes the identity part.
4. While training wide residual network (network with more filters in each layer but shallow depth), found that it leads to poor results because of poor quality of learning due to too many parameters.
5. Resnet validation loss diverges from training loss after a few epochs so we need a method of strong regularization so that both are as close as possible while reducing the training loss.

**Some experiments we didn't do that should have been done?**

1. Fine tuning a pre-trained resnet model on Imagenet dataset. (higher chances of getting results)
2. Dropout
3. Equal proportion of images while training so that objective function in a batch is not biased towards any one class.
4. Boosting the model to classify better when images have approximately equal probability(40-60%) of falling in either class.