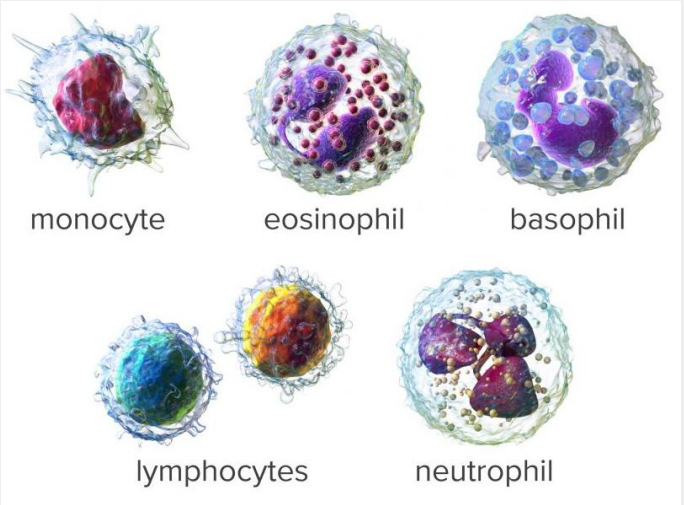
Instance Segmentation of Peripheral Blood Smear and Refining Classification via Domain Adaptation

Jimut Bahan Pal
(B1930050)
RKMVERI

Under the Guidance of
Br. Tamal Maharaj

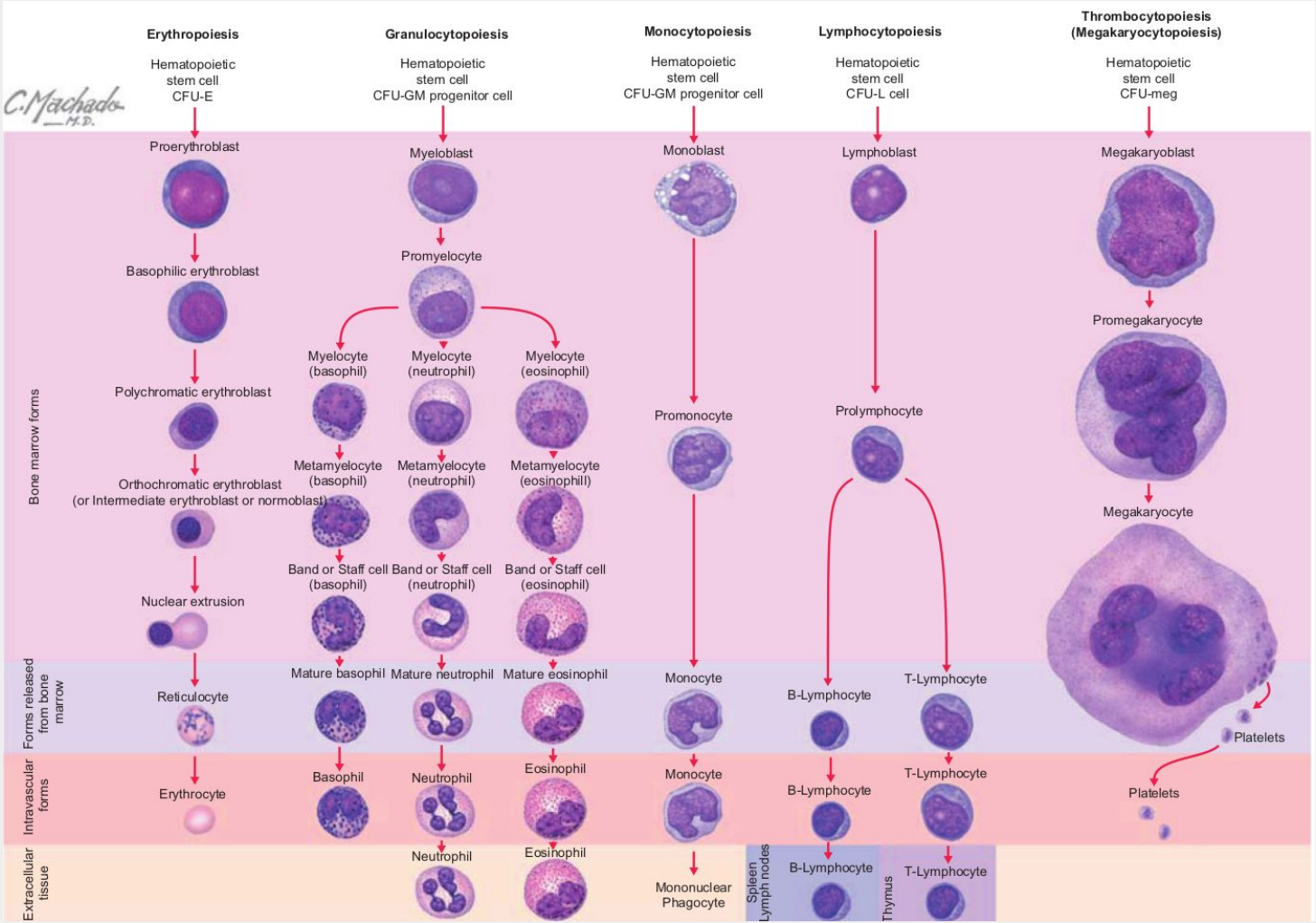
Problem: Find instances of White Blood Cells (WBC) from **peripheral blood smear** and detect which classes it belongs to.

- Main types of WBC's are shown in the figure (highly illustrative).
- **Peripheral blood smear:** A procedure to count and investigate blood samples under a **microscope**.
- Subtask: **Classify, Detect and Segment data.**
- This is a difficult task, since, **getting data** is **costly**. **Data annotation** is done by **medical experts** which is even **costlier**.

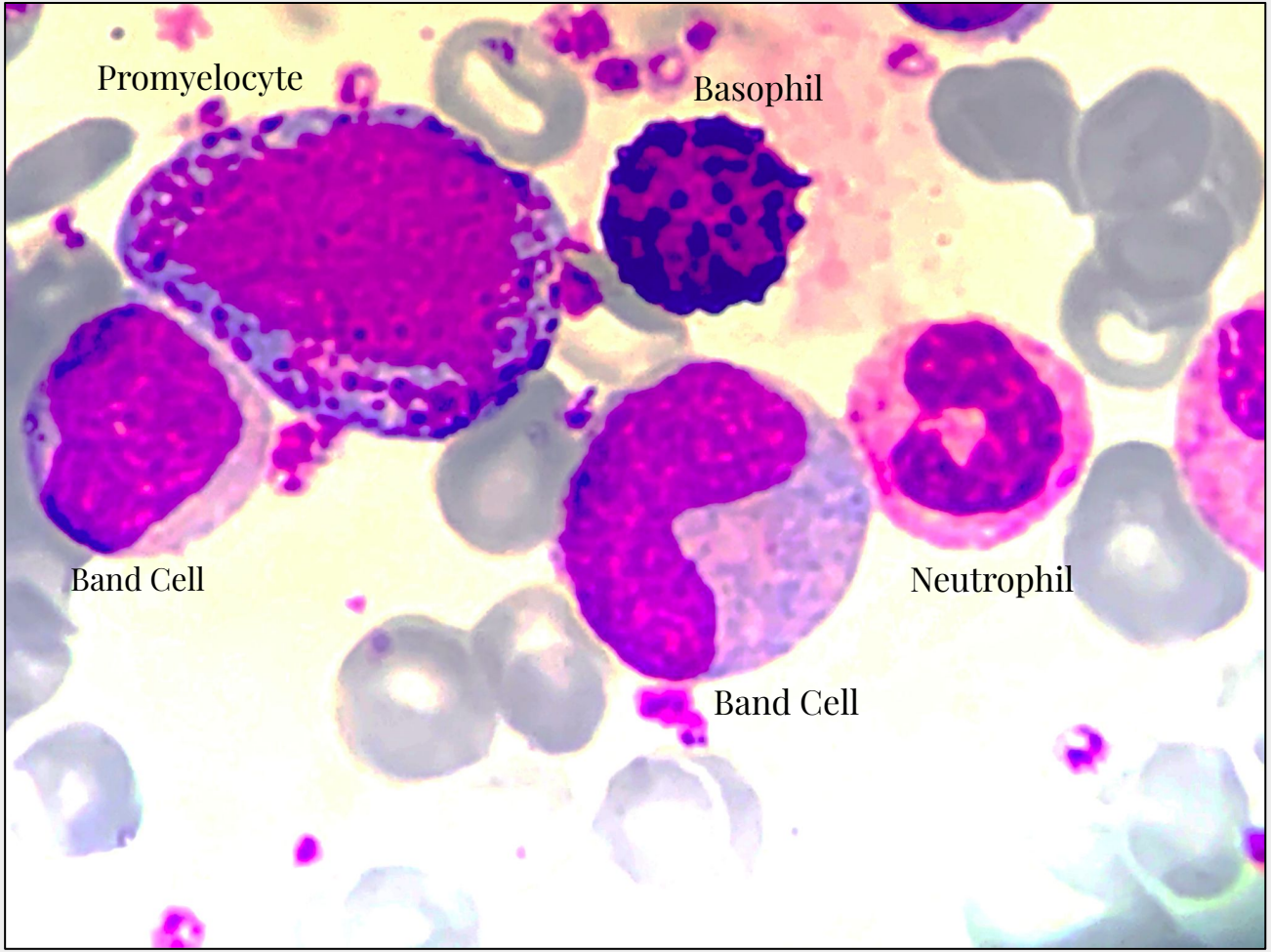


WBC's main types (Picture Courtesy: News)
https://centralalabamawellness.org/wp-content/uploads/2020/01/an-infographic-showing-the-different-types-of-white-blood-cells.jpg

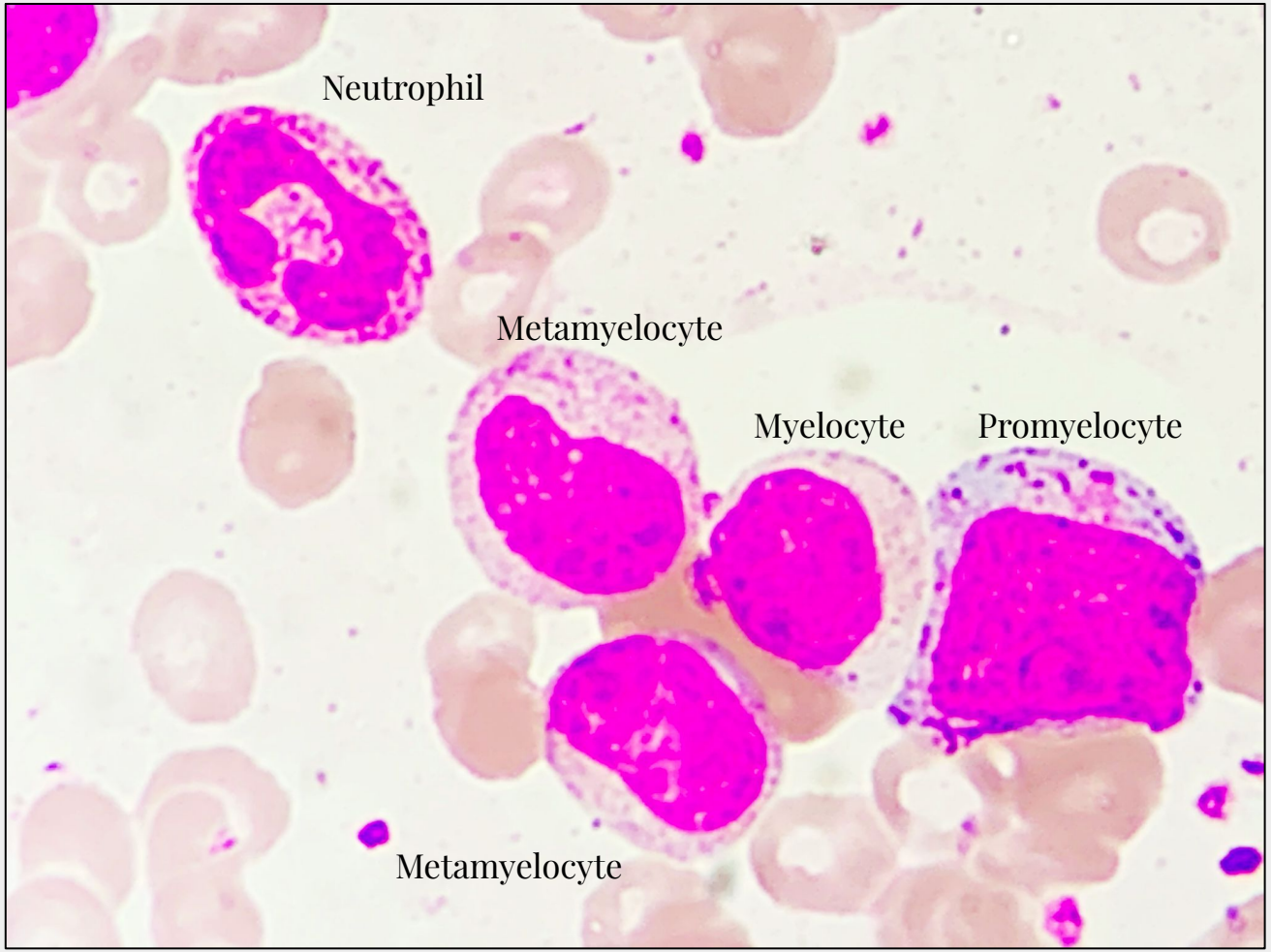
- Different types of hematopoietic cells present in humans are shown in right.
- We are interested in a very small class of this superset of cell classes.
- Our samples of data is much similar to this, but it may vary in colour.
- Samples are shown in next slides.



Picture Courtesy: Elsevier Inc https://www.netterimages.com/



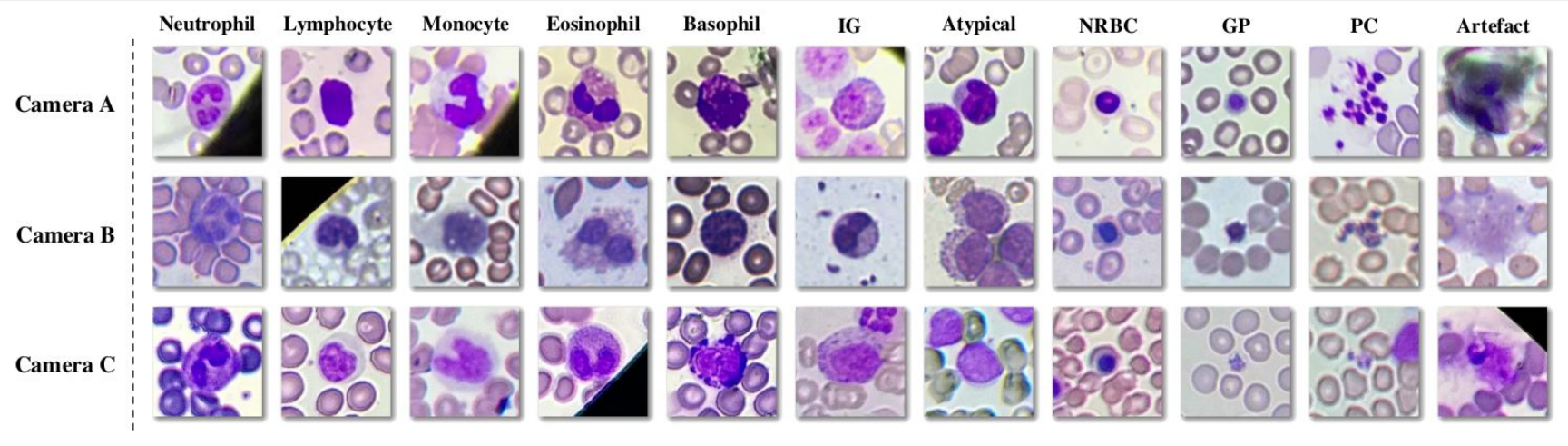
Peripheral Blood Smear - 1



Peripheral Blood Smear - 2

Related Work

- Unsupervised Domain Adaptation on a few proprietary datasets.
- They proposed techniques which solves the problem of classification when images are captured from different cameras.
- Differences in Camera/Microscope types, lenses and lighting conditions etc.



*Prashant Pandey, Prathosh AP, Vinay Kyatham, Deepak Mishra, and Tathagato Rai Dastidar. **Target-independent domain adaptation for WBC classification using generative latent search**. CoRR, abs/2005.05432, 2020. URL https://arxiv.org/abs/2005.05432.

Related Work

- They have introduced the **PBC normal DIB** dataset.
- The dataset have single cell per slide.
- There are 8 classes with sub-classes as discussed later
- Compared the classification accuracy of VGG-16 and Inception V3 model.

*Andrea Acevedo, Santiago Alférez, Anna Merino, Laura Puigví, and José Rodellar. **Recognition of peripheral blood cell images using convolutional neural networks.** Comput. Methods Programs Biomed., 180, 2019. doi: 10.1016/j.cmpb.2019.105020. URL https://doi.org/10.1016/j.cmpb.2019.105020.

Related Work

- Self supervised techniques to extract WBC from slides.
- The first module extract the foreground region by k-means clustering and generates a coarse WBC region by concavity analysis.
- The second module uses an SVM classifier to extract the cells.
- This is essentially a work for extraction of WBCs.

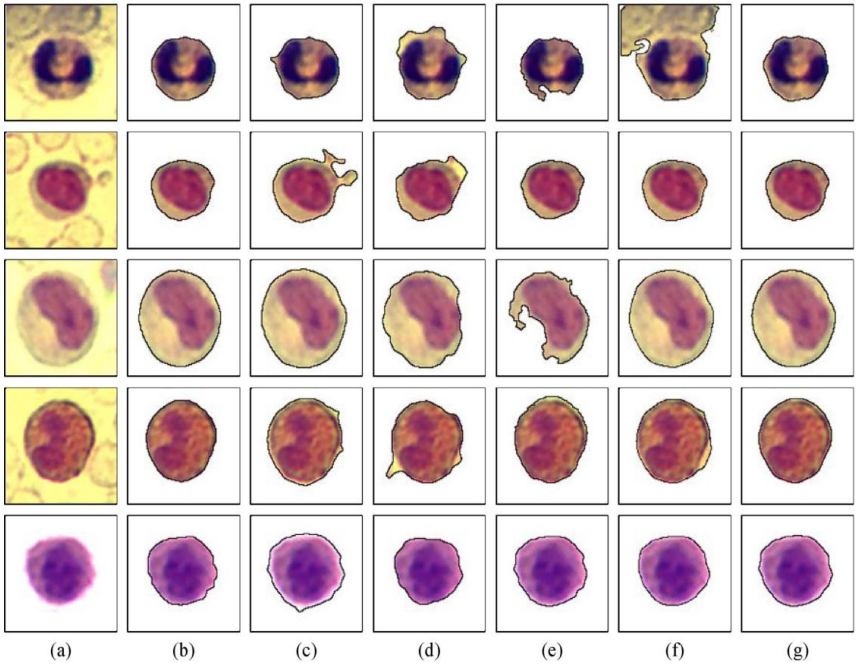


Fig. 13. Example segmentation results on Dataset 1. From top to bottom: neutrophils, lymphocytes, monocytes, eosinophils and basophils. (a) The original sub-images containing different types of WBCs. (b) The ground truth contours. (c) Results of FCM. (d) Results of CGS. (e) Results of SVA. (f) Results of U-Net. (g) Results of our self-supervised learning approach.

*Xin Zheng, Yong Wang, Guoyou Wang, and Jianguo Liu. **Fast and robust segmentation of white blood cell images by self-supervised learning**. Micron, 107:55–71, 2018.

Related Work

- Introduced the Raabin WBC dataset containing 40,000 images of WBCs.
- Accurate and early detection of anomalies in Peripheral WBC's plays a crucial role in diagnosing hematologic diseases.
- 1145 cells have nucleus and cytoplasm extracted.
- The authors have compared the accuracy of different state-of-the-art backbone models and showed their result.

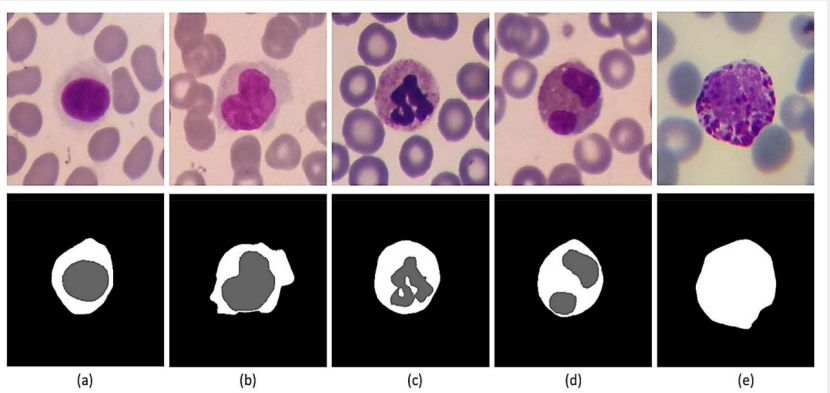
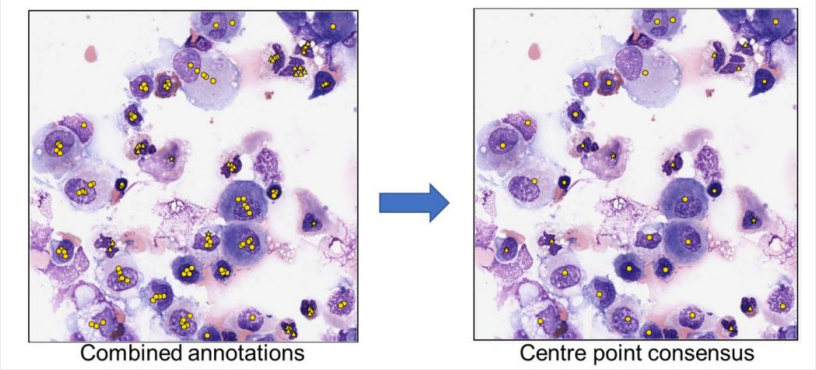


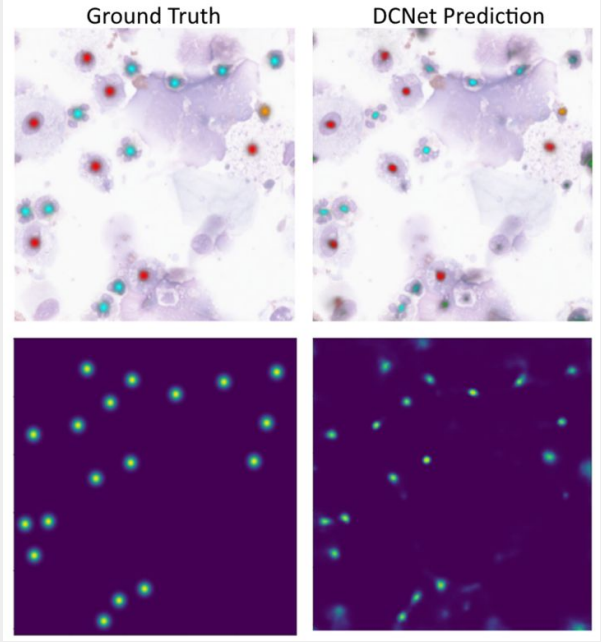
Figure 9. Some samples of ground truths provided in the Raabin-WBC dataset. First row contains the original cropped images of white blood cells. Second row contains the ground truths of some nuclei and cytoplasm. The columns (**a**), (**b**), (**c**), (**d**), and (**e**) show lymphocyte, monocyte, neutrophil, eosinophil, and basophil,

*Kouzehkanan, Z. M., Saghari, S., Tavakoli, S., Rostami, P., Abaszadeh, M., Mirzadeh, F., Satlsar, E. S., Gheidishahran, M., Gorgi, F., Mohammadi, S., & Hosseini, R. (2022). A large dataset of white blood cells containing cell locations and types, along with segmented nuclei and cytoplasm. *Scientific reports*, *12*(1), 1123. https://doi.org/10.1038/s41598-021-04426-x

Related Work

- Researchers have used a novel 1K dataset having different point annotations for cells.
- They have used an U-Net like architecture called as DCNet (Differential Count Network) which detects the cells.
- This helps in finding the cell types by just using point detection network.





*Andrea Acevedo, Santiago Alférez, Anna Merino, Laura Puigví, and José Rodellar. **Recognition of peripheral blood cell images using convolutional neural networks.** Comput. Methods Programs Biomed., 180, 2019. doi: 10.1016/j.cmpb.2019.105020. URL https://doi.org/10.1016/j.cmpb.2019.105020.

Basic Summary of our work

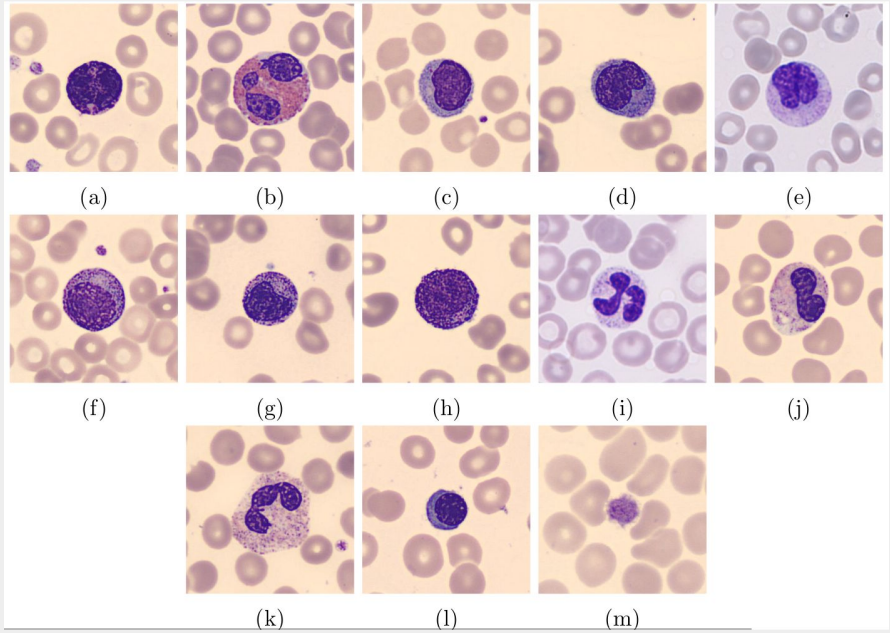
- For extracting the cells, we have performed certain experimental techniques, and selected the best one - **Mask RCNN**.
- Since data is limited, we need to utilize a **similar dataset** which can enhance the detection of the novel dataset which we are going to use.
- For achieving the task, we use **Domain Adaptation** (introduced later).
- The pipeline is simple:
    - First - extract the exact **masks** of the cells using **Mask RCNN**
    - Second - Use a classification model to classify all the **10** classes of the cropped dataset
    - Third - Use the domain adaptation pipeline to refine the classification on the **8** common classes of individual datasets.
- For performing Domain Adaptation, we have found **PBC normal DIB dataset,** which is modified for our specific needs.

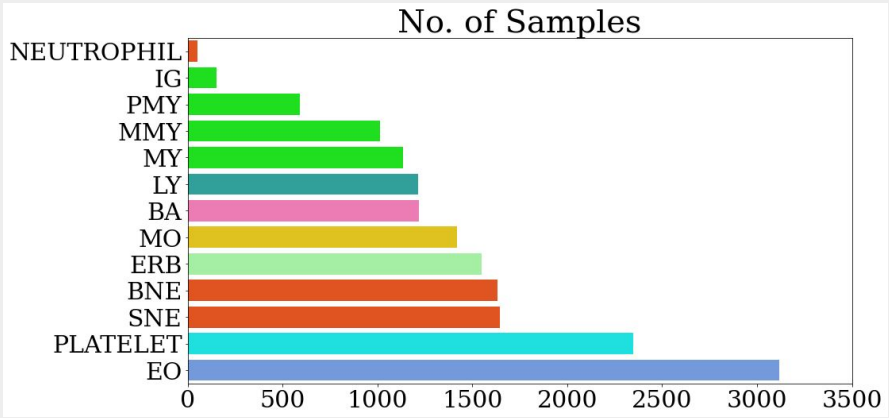
Datasets Used

Datasets used

PBC dataset normal DIB dataset

- There are a total of **17092** images present in this dataset.
- Classes are (top left to bottom right):
  - (a) **Basophil (BA)**
  - (b) **Eosinophil (EO)**
  - (c) **Lymphocyte (LY)**
  - (d) **Monocyte (MO)**
  - **Immature Granulocytes (IG)**
    - (e) Immature Granulocytes (IG)
    - (f) Myelocyte (MY)
    - (g) Metamyelocyte (MMY)
    - (h) Promyelocytes (PMY)
  - **Neutrophil (NEUTROPHIL)**
    - (i) Neutrophil (NEUTROPHIL)
    - (j) Band Neutrophils (BNE)
    - (k) Segmented Neutrophils (SNE)
  - (l) **Erythroblast (ERB)**
  - (m) **Platelet (PLATELET)**





Creation of PBC dataset normal DIB Cropped dataset

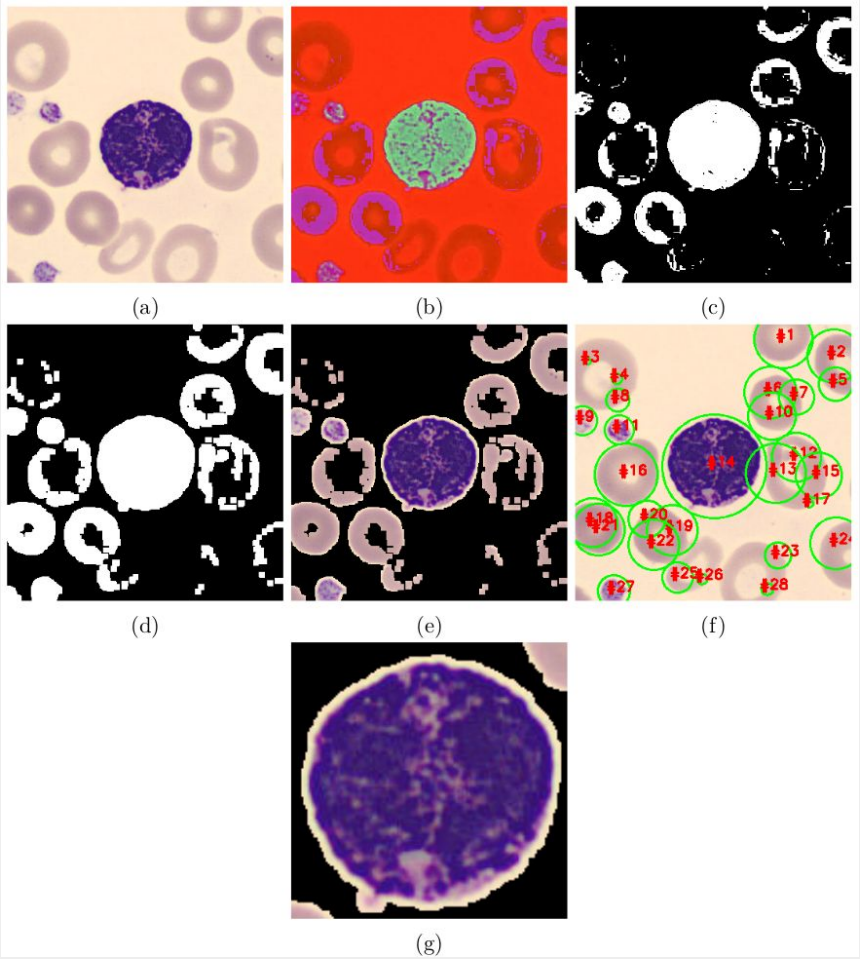
- To perform domain adaptation, we need to **crop out** cell images from the PBC slides, since we are interested in masked cells.
- Algorithm for cropping out is present in the right.
- We use **Watershed algorithm** for extracting the markers and automating the extraction of cells.
- The cells are generally present in the center for the PBC dataset, hence we use this information to select the mask which is present in the center, and crop out accordingly.

Algorithm 1: Algorithm for extracting the cells from **PBC_dataset_normal_DIB** dataset.

for *each* ***image*** *of the* ***PBC_dataset_normal_DIB*** *dataset* **do**
  image ← individual image of **PBC_dataset_normal_DIB** dataset;
  img ← *copy* of **image**;
  hsv ← *convert* **img** from *BGR* to *HSV*;
  lower_blue ← **(100, 20, 20)**;
  upper_blue ← **(300, 245, 245)**;
  mask ← *threshold* **hsv** within [**lower_blue**, **upper_blue**];
  mask ← *dilate* **mask** with 3 iterations;
  res ← *bitwise and* with **img** and **mask**;
  local_Max ← compute the exact *euclidean distance* from every binary pixel to the nearest zero pixel, then find peaks in this distance map with min distance of 20 on **mask**;
  markers ← perform a *connected component* analysis on the local peaks, using $8 - connectivity$, then appy the *watershed algorithm* using **local_Max**;
  labels ← apply *watershed algorithm* using **markers** and **mask**;
  image ← *draw circles* from the uniquely identified contours and *find* the coordinates of the biggest contour and store the **radius** using **labels** on **image**;
  crop ← Take **H/2**, **W/2** as the central point and *crop* out the masked **image** by taking the **radius** as the boundary;
end

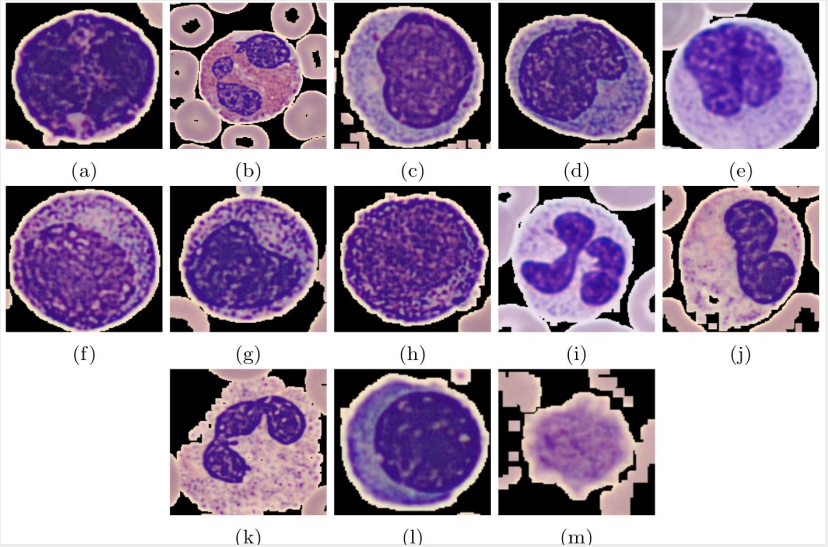
Creation of PBC dataset normal DIB Cropped dataset

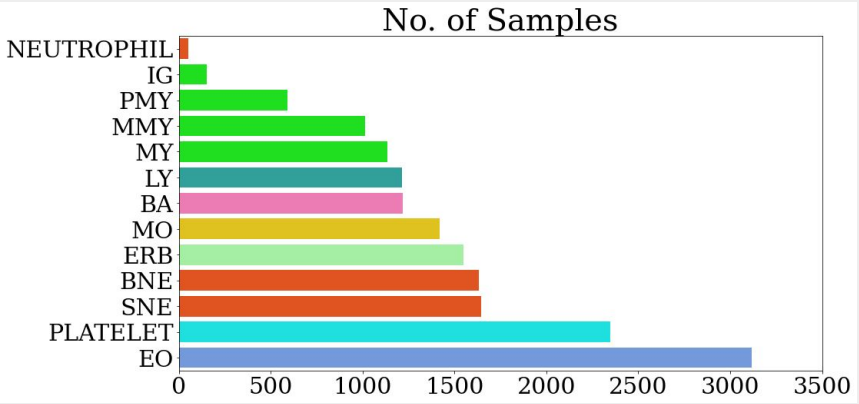
- The methods are shown in the figure.
- From top left: (a) A sample image from PBC normal DIB dataset. (b) The HSV converted image. (c) The thresholded image after applying colour threshold on the HSV image.
- (d) The dilated threshold which acts as mask.
- (e) The masked original image
- (f) Finding the globs and selecting the maximum radius of the globs.
- (g) Cropping out the glob from the center with maximum radius.
- It took **0.1448** seconds to process an image on average.



PBC dataset normal DIB Cropped dataset

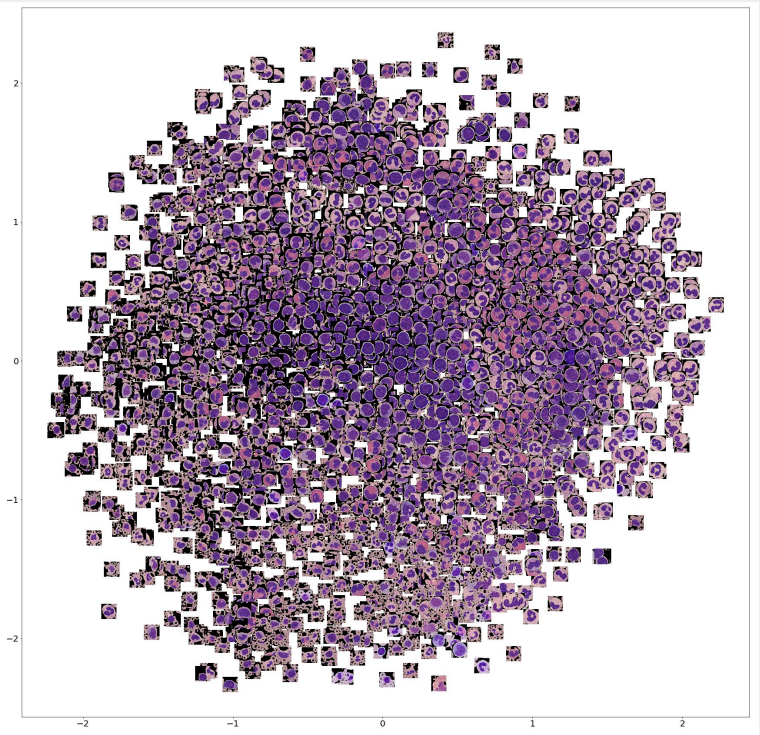
- **PBC dataset normal DIB Cropped dataset**
- There are a total of **17092** images present in this dataset.
- Classes are (top left to bottom right):
  - (a) **Basophil (BA)**
  - (b) **Eosinophil (EO)**
  - (c) **Lymphocyte (LY)**
  - (d) **Monocyte (MO)**
  - **Immature Granulocytes (IG)**
    - (e) Immature Granulocytes (IG)
    - (f) Myelocyte (MY)
    - (g) Metamyelocyte (MMY)
    - (h) Promyelocytes (PMY)
  - **Neutrophil (NEUTROPHIL)**
    - (i) Neutrophil (NEUTROPHIL)
    - (j) Band Neutrophils (BNE)
    - (k) Segmented Neutrophils (SNE)
  - (l) **Erythroblast (ERB)**
  - (m) **Platelet (PLATELET)**





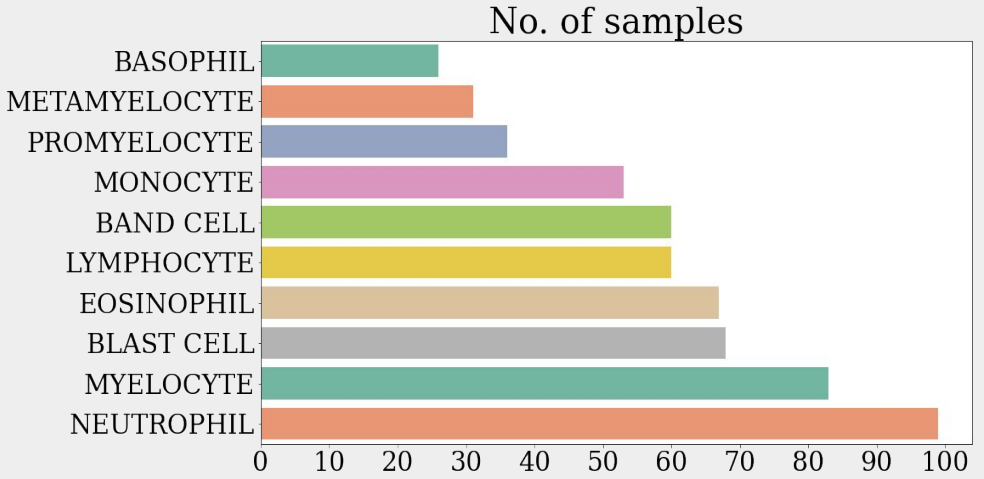
Some Insights on PBC dataset normal DIB dataset cropped

- Visualizing the test set (**2609 images**) using t-SNE (with perplexity **50**)
- Taking PCA (Principal Component Analysis) on **200** components of the **360x360x3** images



Smear Slides (Novel) dataset

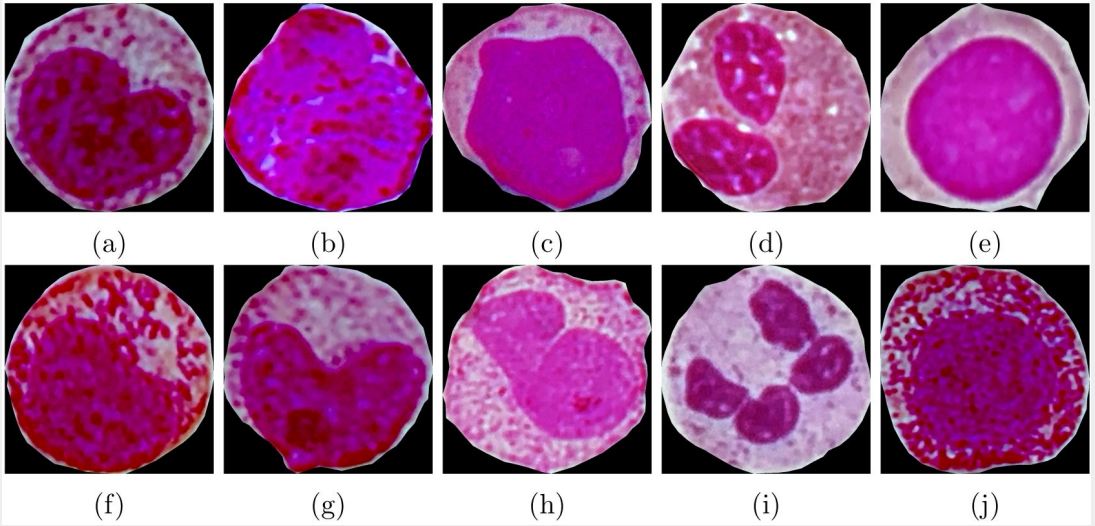
- We have annotated **583** images with instance mask using CVAT (Computer Vision Annotation Tool) in yml file format.
- A sample of annotated image from the Basophil class is shown in right.
- This format is necessary for getting the **mask** using **Mask RCNN**, which converts **yml** to images internally for **instance segmentation**.
- The classes that are present are:
  - Band Cell
  - Basophil
  - Blast Cell
  - Eosinophils
  - Lymphocytes
  - Myelocytes
  - Metamyelocytes
  - Monocytes
  - Neutrophil
  - Promyelocytes



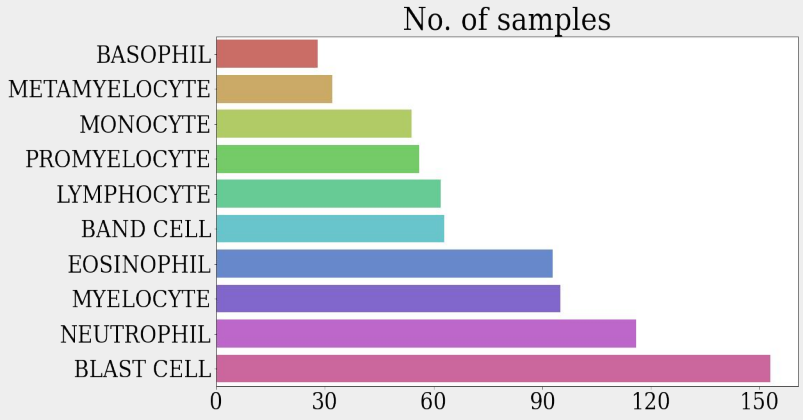
Smear Slides Cropped dataset

- There are a total of **755** images present in this dataset.

- Classes are (top left to bottom right):
  - (a) Band Cell
  - (b) Basophil
  - (c) Blast Cell
  - (d) Eosinophils
  - (e) Lymphocytes
  - (f) Myelocytes
  - (g) Metamyelocytes
  - (h) Monocytes
  - (i) Neutrophil
  - (j) Promyelocytes

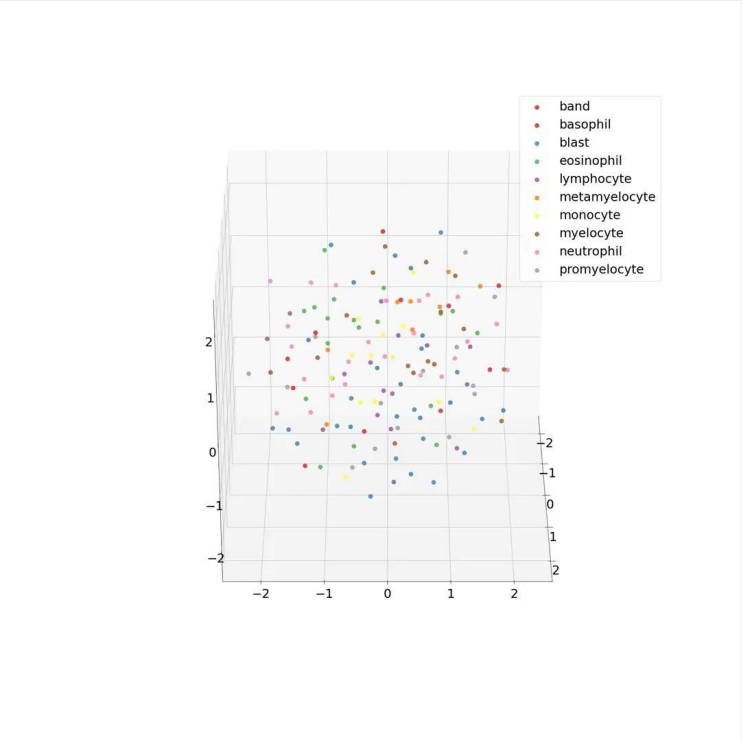


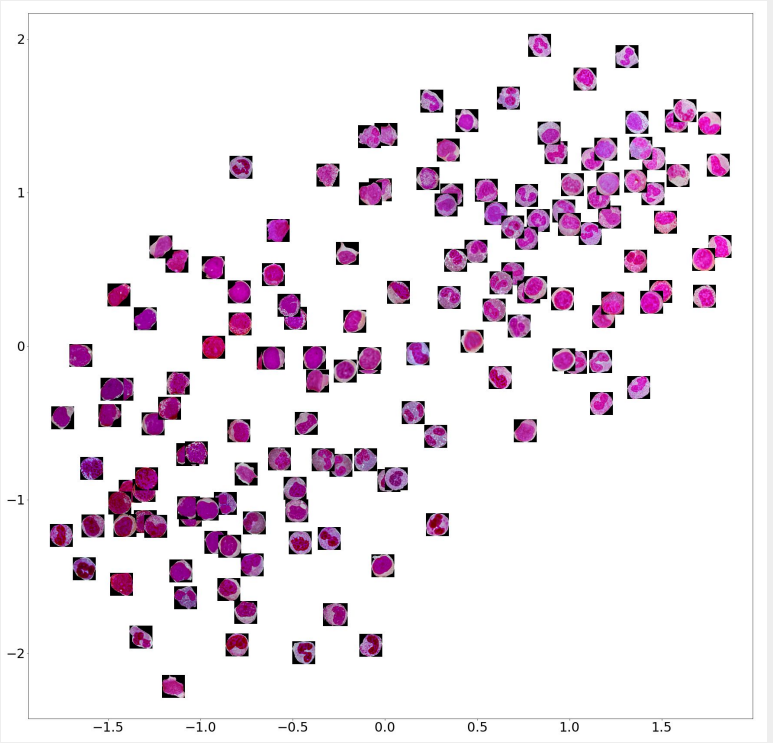
- The distribution of the Smear Slides Cropped dataset is shown below (a few images more than individual classes present in the segmentation dataset).



Some Insights on Smear Slides Cropped dataset

- Visualizing the test set (**151 images**) using t-SNE (with perplexity **30**)
- Taking PCA on **50** components of the **360x360x3** images

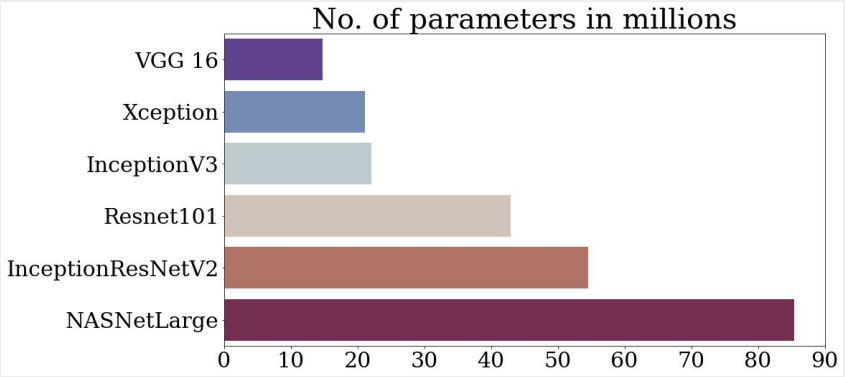


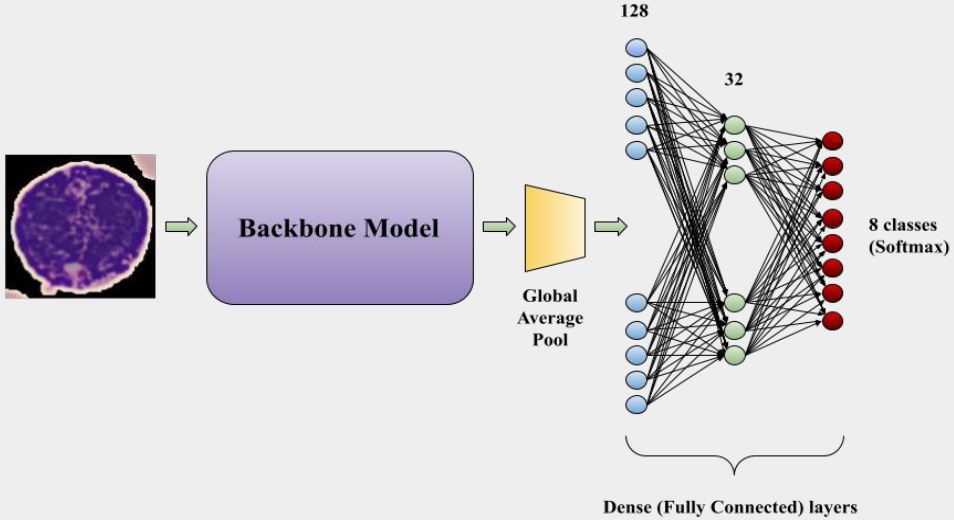


Classification Models

Classification model applied on PBC datasets

- The following architecture shows the use of **backbone model** for selecting the best model; initially trained on ImageNet dataset.
- The **bar plot** shown in left is the comparison of **model parameters** (in million).
- We have used a seed, and divided the dataset into **64-16-20,** as train-validation-test split.
- **Adam optimizer** with a learning rate of **1e-5** (batch-size = **16**) is used, **Categorical Cross Entropy** was used as loss function.





Some Metrics

Here are some of the metrics that are used during training of the model :

$True\ Positive\ (TP)$ The model correctly predicts a positive class

$True\ Negative\ (TN)$ The model correctly predicts a negative class

$False\ Positive\ (FP)$ The model predicts positive class when it is a negative class

$False\ Negative\ (FN)$ The model predicts negative class when it is a positive class

$$Accuracy\ (AC) = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Recall\ or\ Sensitivity\ (SE) = \frac{TP}{TP + FN}$$

$$Precision\ (PC) = \frac{TP}{TP + FP}$$

$$F1 - score = \frac{2\ (PC \times SE)}{PC + SE}$$

Some Metrics

- Categorical Cross-Entropy Loss function may be written as follows

$$L_{y'}(y) = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i' \log_e(y_i) + \left(1 - y_i'\right)\log_e\left(1 - y_i\right)\right)$$

- Binary Cross-Entropy Loss function may be written as follows

$$L_y(y') = -\frac{1}{N}\left(\sum_{i=1}^{N}\left(y_i' \log_e(y_i)\right)\right)$$

Where Predicted class is y' and True value is y

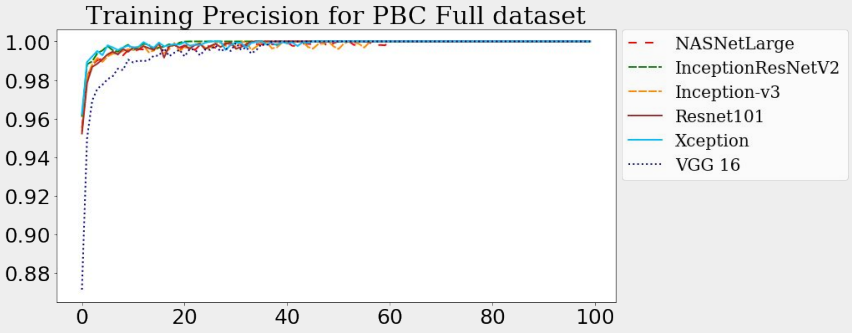
Classification model applied on PBC datasets

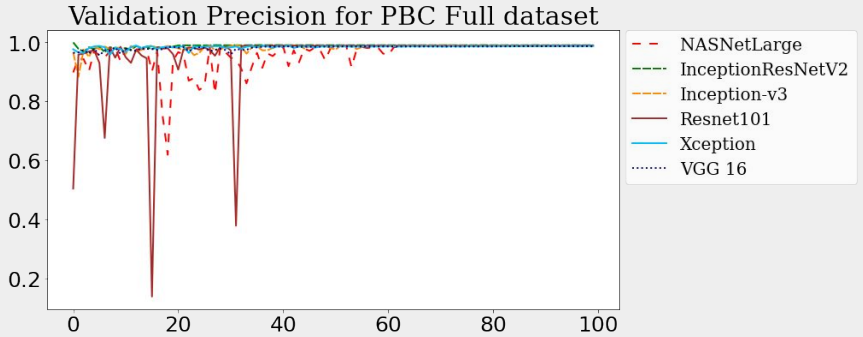
- Various backbone models have been applied to get the best **Precision**, **Recall** and **F1-score** on test set.
- One with keeping the conv layers freezed, only training the fully connected layers and another with fine tuning the whole architecture.
- Fine tuning of architecture gives the best result, and **Xception model** is selected which does particularly well in PBC cropped datasets.

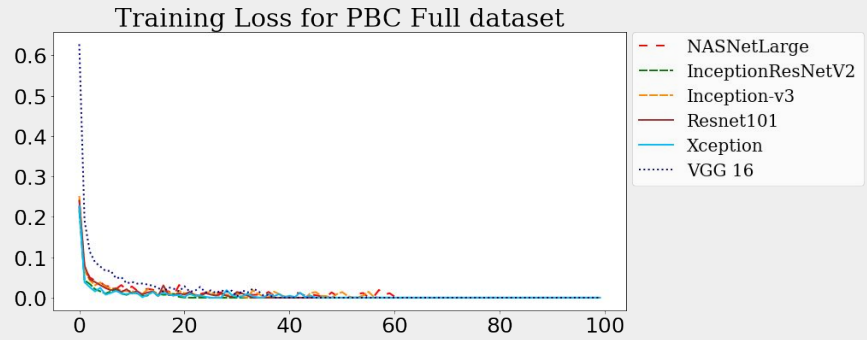
Model Name	Dataset	Type	Precision	Recall	F1-score
Inception V3 [22]	PBC_cropped	fine tuned	96.01	95.74	95.70
	PBC_cropped	freezed	91.15	91.15	90.97
	PBC_dataset	fine tuned	98.73	98.88	98.89
	PBC_dataset	freezed	91.61	91.42	91.28
Inception-ResNetV2 [21]	PBC_cropped	fine tuned	98.84	98.99	98.82
	PBC_cropped	freezed	93.73	93.90	93.66
	PBC_dataset	fine tuned	**99.02**	**99.07**	**98.93**
	PBC_dataset	freezed	93.60	93.41	93.45
NASNetLarge [26]	PBC_cropped	fine tuned	96.63	96.51	96.64
	PBC_cropped	freezed	92.51	92.38	92.39
	PBC_dataset	fine tuned	98.84	98.92	98.75
	PBC_dataset	freezed	93.63	93.67	93.78
VGG16 [16]	PBC_cropped	fine tuned	98.49	98.29	98.22
	PBC_cropped	freezed	95.03	94.85	95.05
	PBC_dataset	fine tuned	98.73	98.78	98.69
	PBC_dataset	freezed	93.60	93.23	93.27
Xception [5]	PBC_cropped	fine tuned	98.96	98.81	98.75
	PBC_cropped	freezed	93.59	93.46	93.29
	PBC_dataset	fine tuned	98.84	98.81	98.75
	PBC_dataset	freezed	91.51	91.19	91.27
Resnet101 [10]	PBC_cropped	fine tuned	98.37	98.56	98.40
	PBC_cropped	freezed	71.05	68.95	68.47
	PBC_dataset	fine tuned	98.84	98.92	98.75
	PBC_dataset	freezed	68.87	70.22	68.03

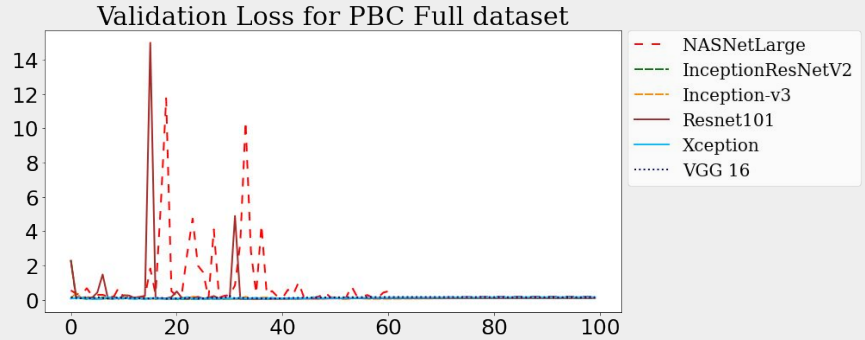
Classification model applied on PBC datasets

- Some metrics are shown, the spikes might be the result of outliers which does not help the optimizer to reach the local minima, for some models.









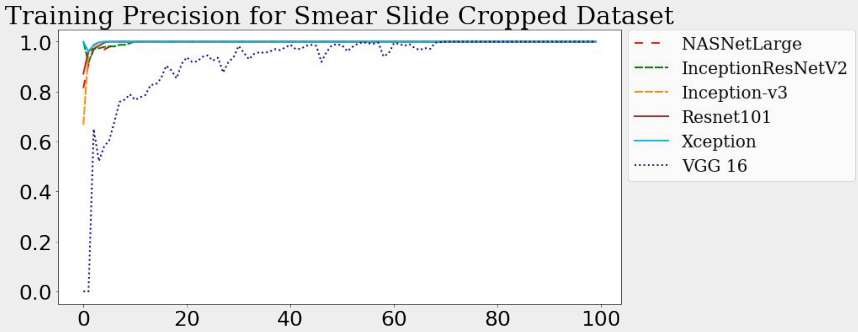
Classification model applied on Smear Slides Cropped dataset

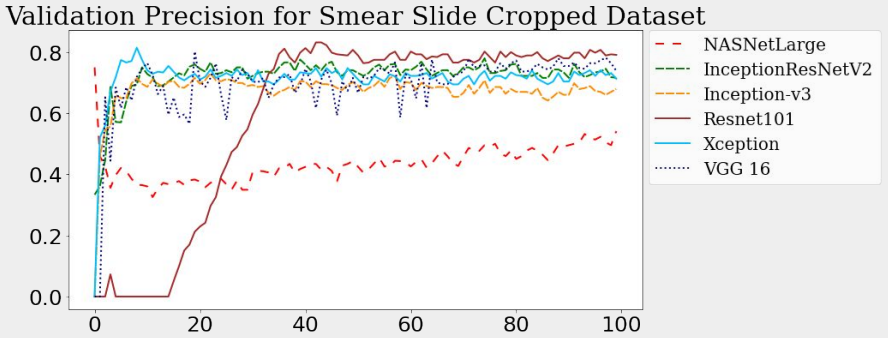
- Similar experiments have been conducted on Smear Slides Cropped dataset (without data augmentation).
- Just fine-tuning was applied, since it is bound to give good results.
- Again, **Xception model** does relatively well in this experiment too.

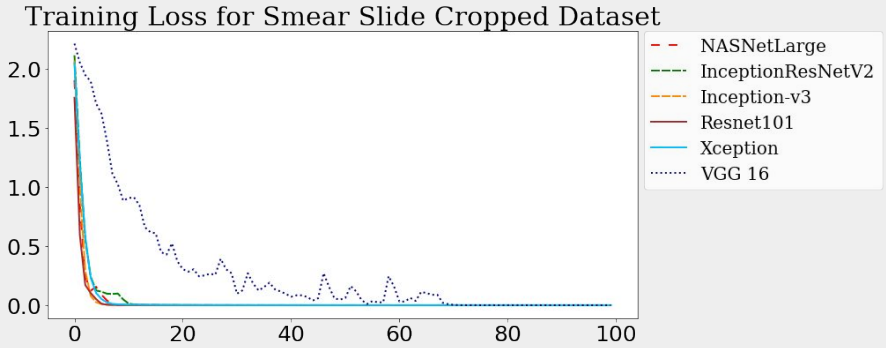
Model Name	Precision	Recall	F1-score
Inception V3	77.41	75.57	74.44
Inception-ResNetV2	81.07	76.91	74.93
NASNetLarge	61.79	54.37	51.29
VGG16	73.54	72.72	72.62
Xception	**82.15**	**78.80**	**76.96**
Resnet101	72.94	76.84	74.39

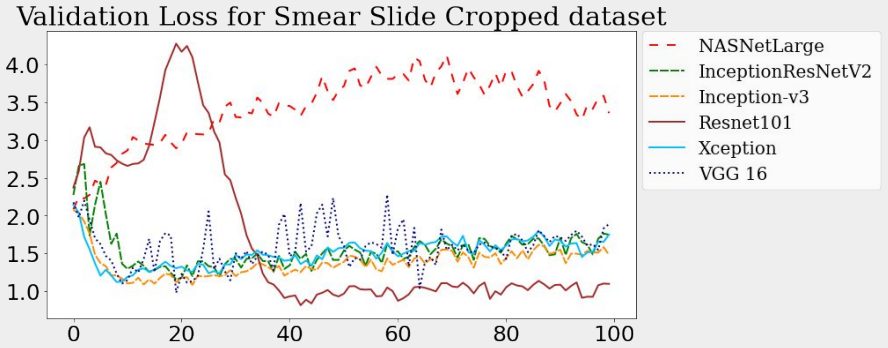
Classification model applied on PBC datasets

- The validation dataset is not able to fully approximate the training dataset, hence the plots seems to be irregular for some models.



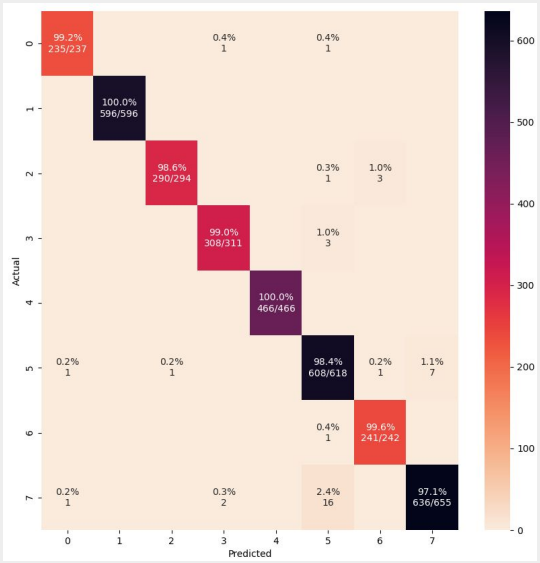


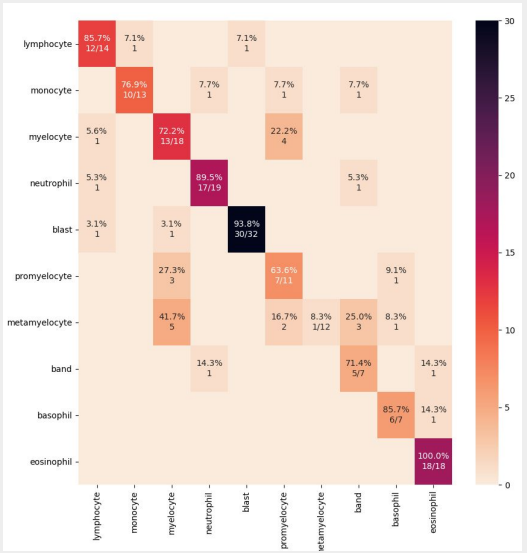




Confusion matrix

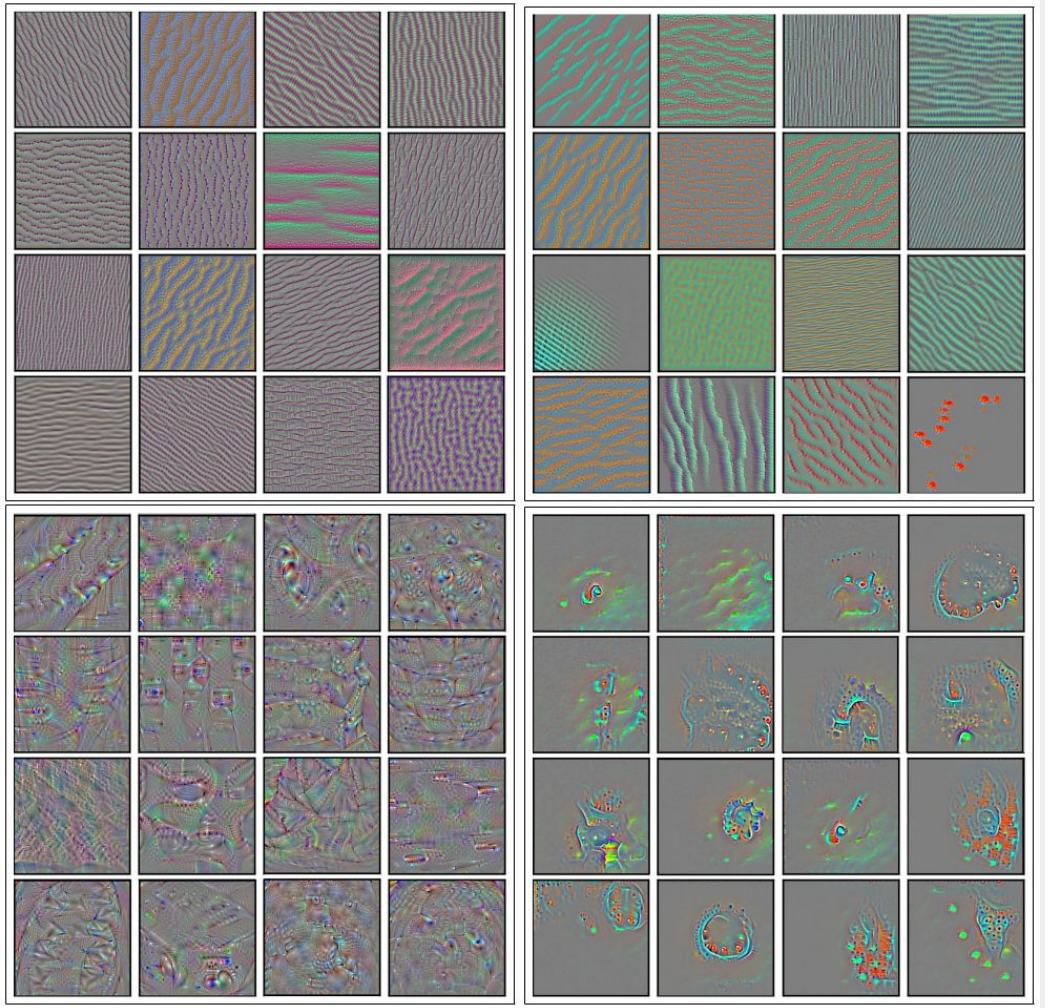
- The confusion matrix obtained by the best model in PBC Cropped (left), and Smear Slides Cropped (right) datasets.
- PBC dataset was performing better in test set due to large number of data.





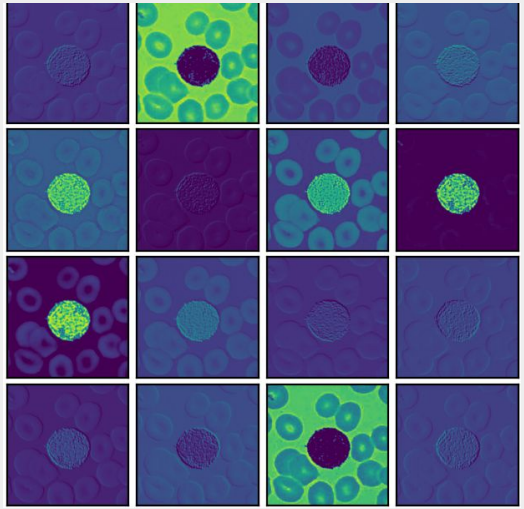
Visualizing what Convnet learns

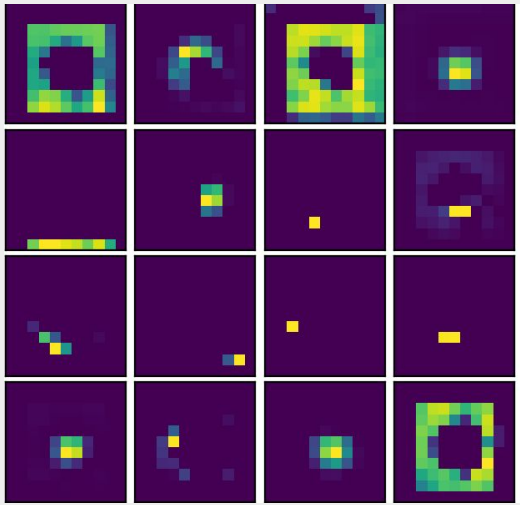
- For spotting the difference between the freezed and fine-tuned, we present the features which the model learns to respond to (for VGG16 model).
- The **left** part shows the **freezed**, hence the conv layers are not trained, the initial model was trained on ImageNet dataset, and the patterns are related to natural images.
- The upper layer shows some of the initial layers (**block2_conv2**), and lower layer shows the final convolutional layers (**block5_conv3**).
- The fine-tuned blocks learns **problem specific textures** (trained on PBC datasets), whereas the final layers learns the cellular structures only.



Classification model applied on PBC datasets

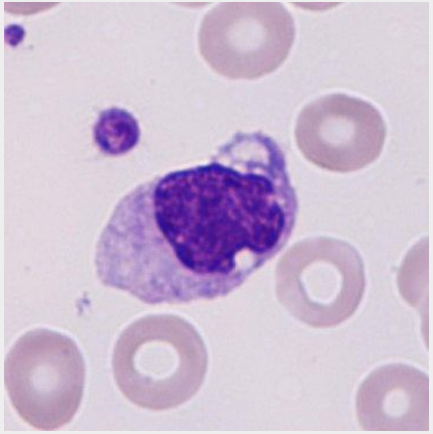
- The feature maps that were generated by the initial and the final convolutional layer of VGG 16.
- The initial layers represents features which are more image oriented, the final layers generate features which are more vector oriented.
- There are a total of 64 feature present in the initial layer, whereas 512 feature present in the final layer.
- Each **filter** gives a **different feature**, so the number of **features increases with filter maps**.

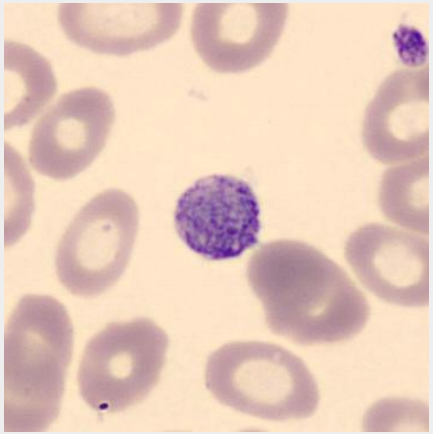




Experimental Methods for Masking

- In the initial stage of this project, we had very less data, so we were experimenting with classical methods for detecting and localizing the cells which are not so data driven.
- These methods are very experimental and will not do much good in real environment.
- We were mainly using Watershed Algorithm, some colour thresholding for blob detectors and other experiments to see how they perform.





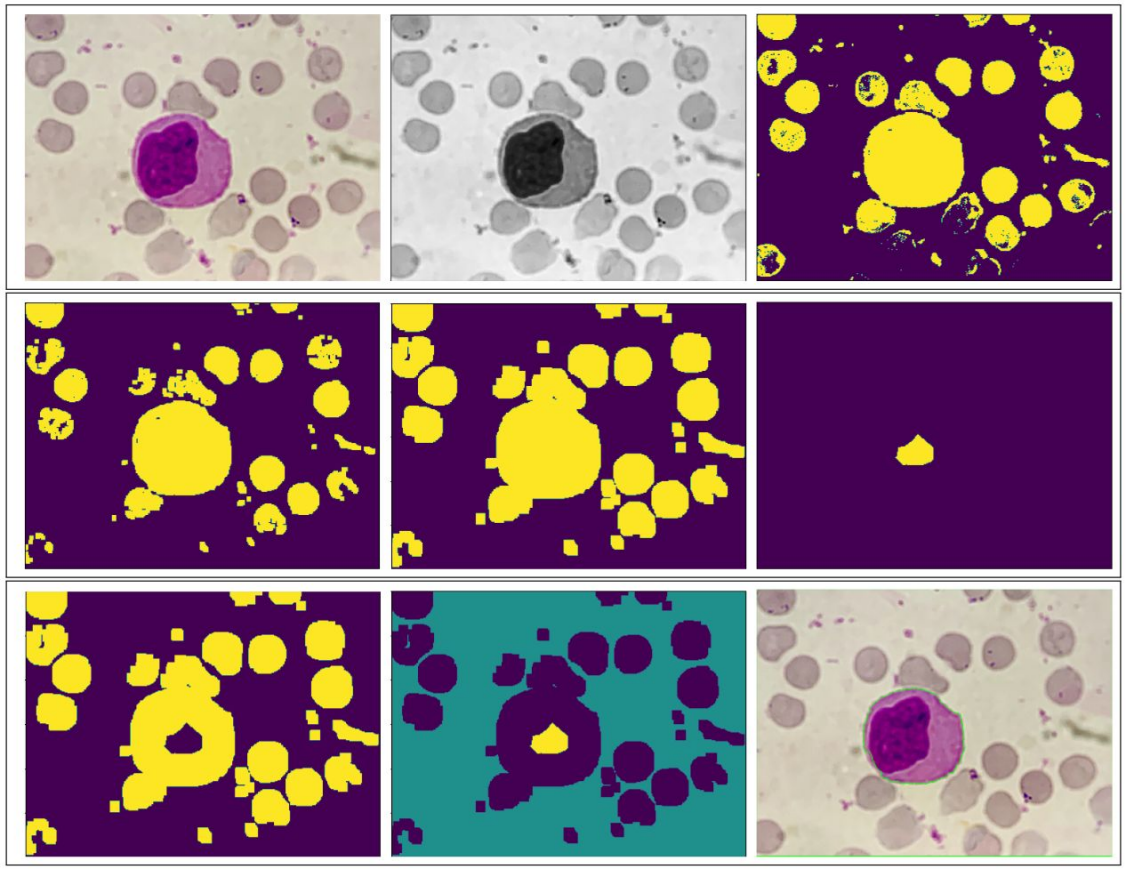
Samples from the PBC dataset

Segmentation techniques used

1. **Watershed Algorithm**
2. DoH, DoG and LoG
3. Grad CAM based Novel method
4. Mask RCNN

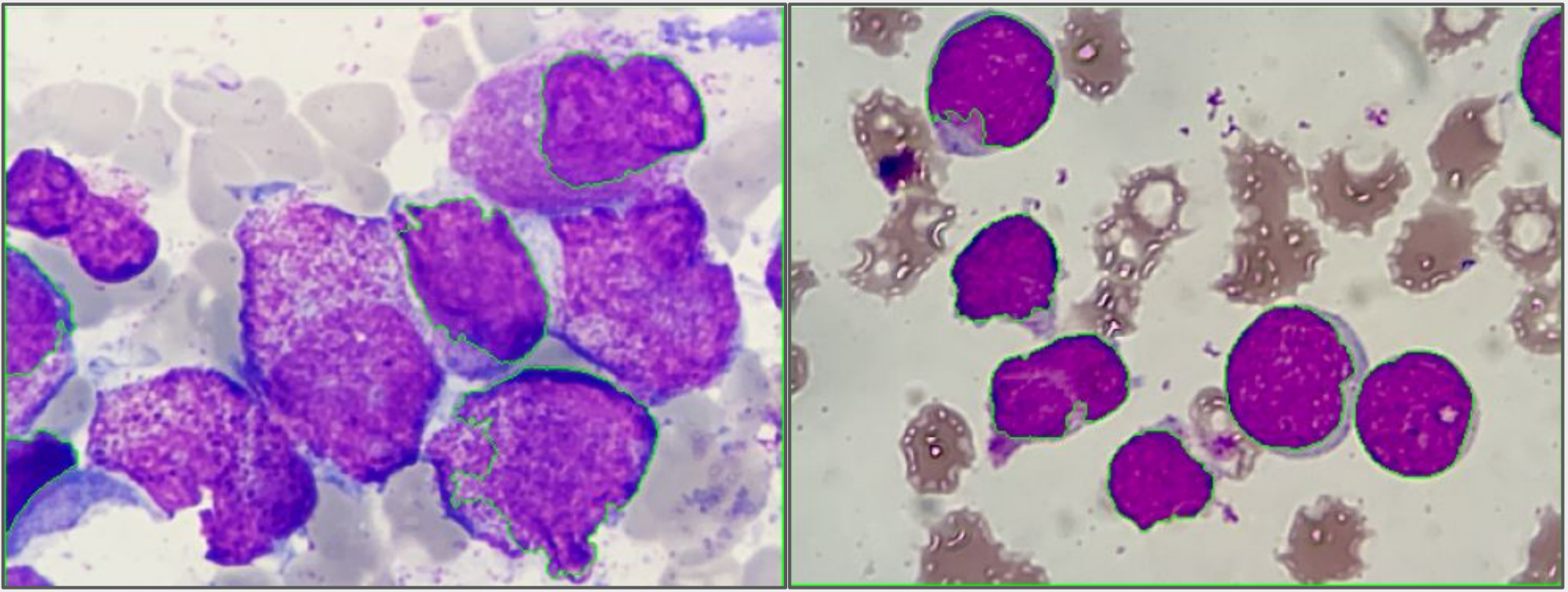
Watershed Algorithm

- Watershed is a classical algorithm for image segmentation.
- Using watershed one can use user defined **markers**, to fill **local minima of topographic elevations.**
- It treats input images' **pixel's intensity values as topographic elevations**.
- The filling starts from **local minima** and continues till there is a **boundary** between two **water bodies** (here, also referred to as images)
- Figure in right shows the different steps in performing Watershed Segmentation in a Smear Slide image.



Watershed Algorithm - Results

- Left: A bad example, Right: A relatively good example for performing watershed for segmenting out the cells.
- Hence watershed will be **ineffective** in this case, when we are dealing with high precision.

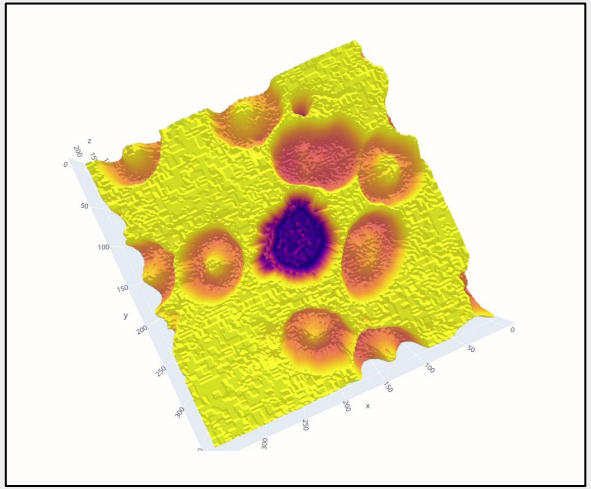


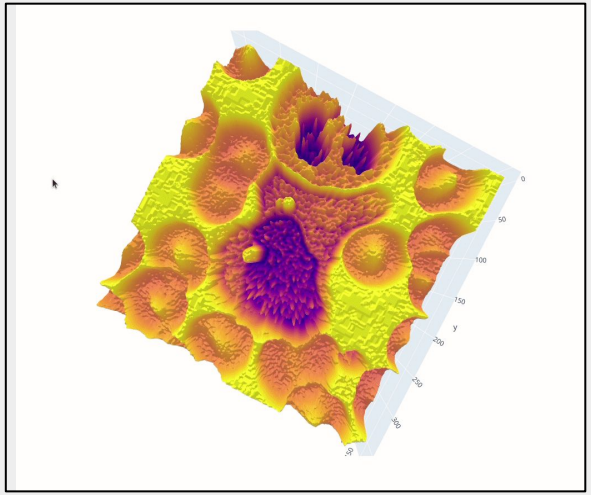
Segmentation techniques used

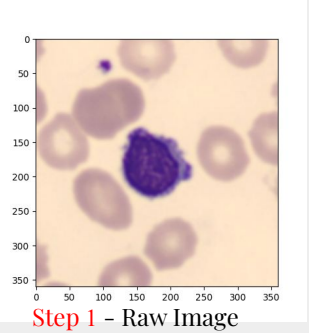
1. Watershed Algorithm
2. **HoG, DoG and LoG**
3. Grad CAM based Novel method
4. Mask RCNN

HoG, DoG and LoG

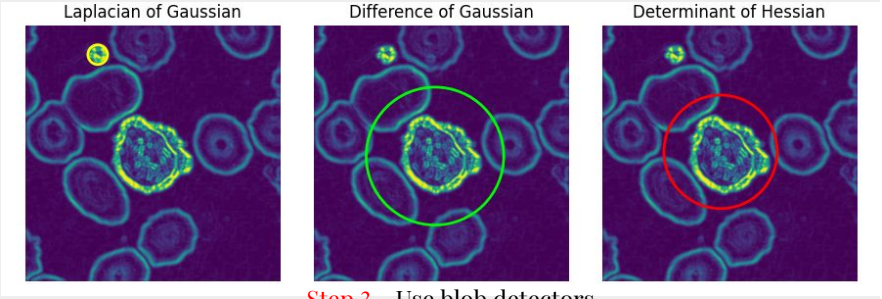
- We are testing the PBC_normal_DIB dataset via classical methods.
- Firstly we need to read a raw image, and convert it to **grayscale**, and apply **sobel filter** on it.
- We select the resulting image and use blob detectors, i.e., **Laplacian of Gaussian** (**LoG**), **Difference of Gaussian** (**DoG**) and **Hessian of Gaussian** (**HoG**).
- We find the blob, and select the **solid areas**, one over the other as a thick layer of confidences.
- We apply **thresholding** and repeated **erosion** to get a good area of localisation.
- We then do **dilation** to get the detected **approximate mask** which can give an average estimation of the cell in the image.



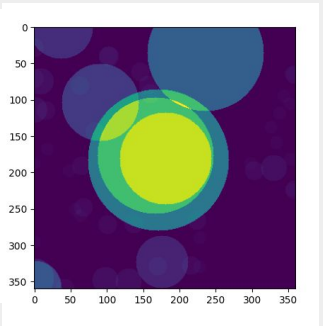




Step 1 - Raw Image

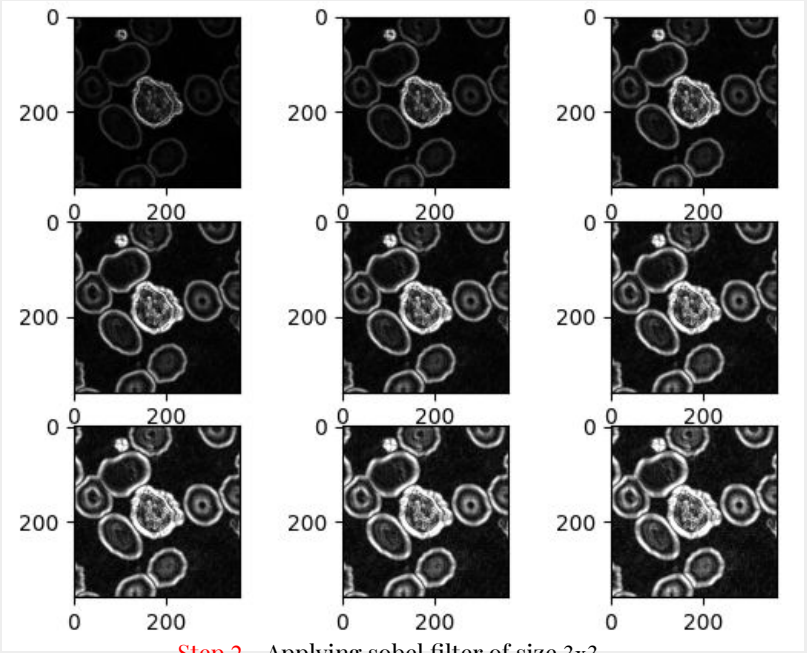


Step 3 - Use blob detectors

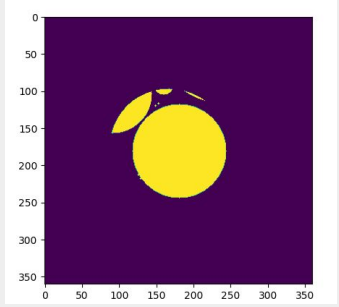


Step 4 - Use blob to make a layer of confidences, a factor is multiplied, bigger the circle, more the value is added.

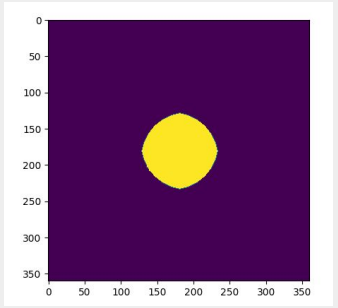
Factor = $e^{\log_2 r}$



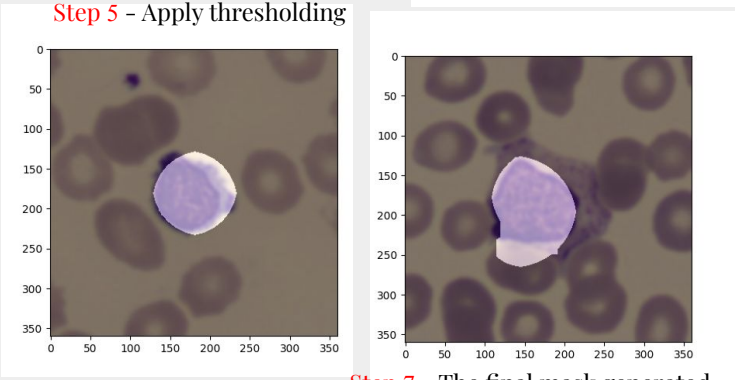
Step 2 - Applying sobel filter of size 3x3



Step 5 - Apply thresholding



Step 6 - Repeated Erosion and then dilation to get approximate area of localisation



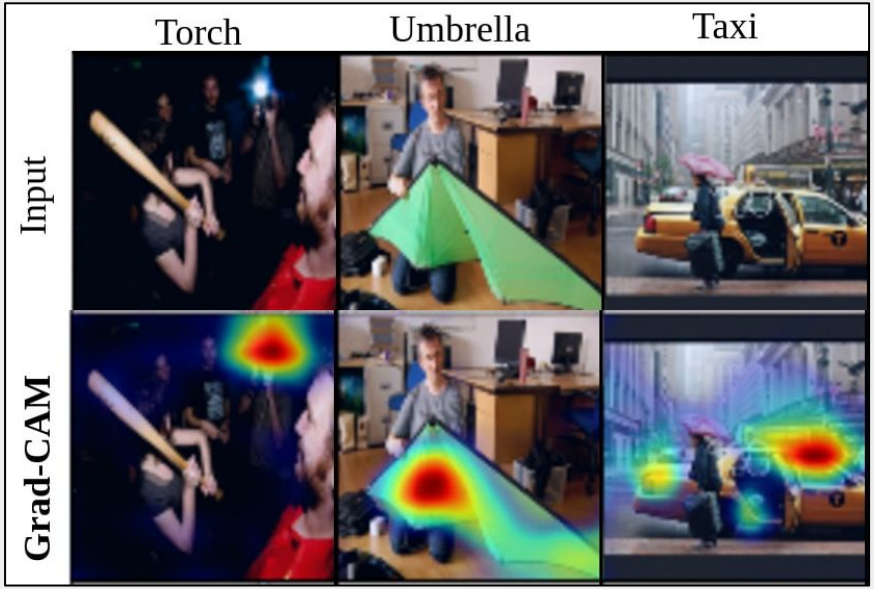
Step 7 - The final mask generated

Segmentation techniques used

1. Watershed Algorithm
2. DoH, DoG and LoG
3. **Grad CAM based Novel method**
4. Mask RCNN

What is **GRAD CAM ?**

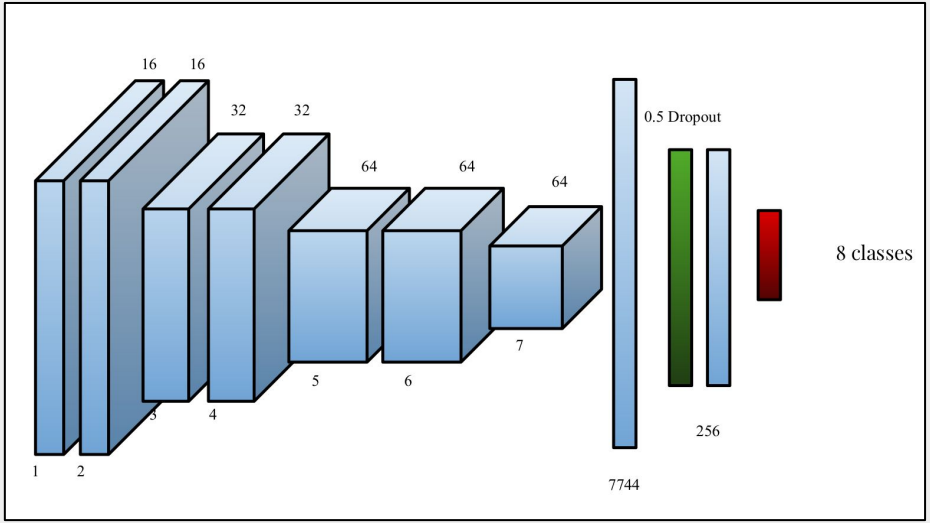
- Visual explanations from deep networks via **gradient localization**.
- **Robust** to **adversarial perturbations**.
- More **faithful** to underlying models.
- Helps to achieve **model generalization** by identifying **underlying bias** and helps us understand what the model **sees**.
- **Deeper layers** captures more **semantic concepts** and they are used for getting better results



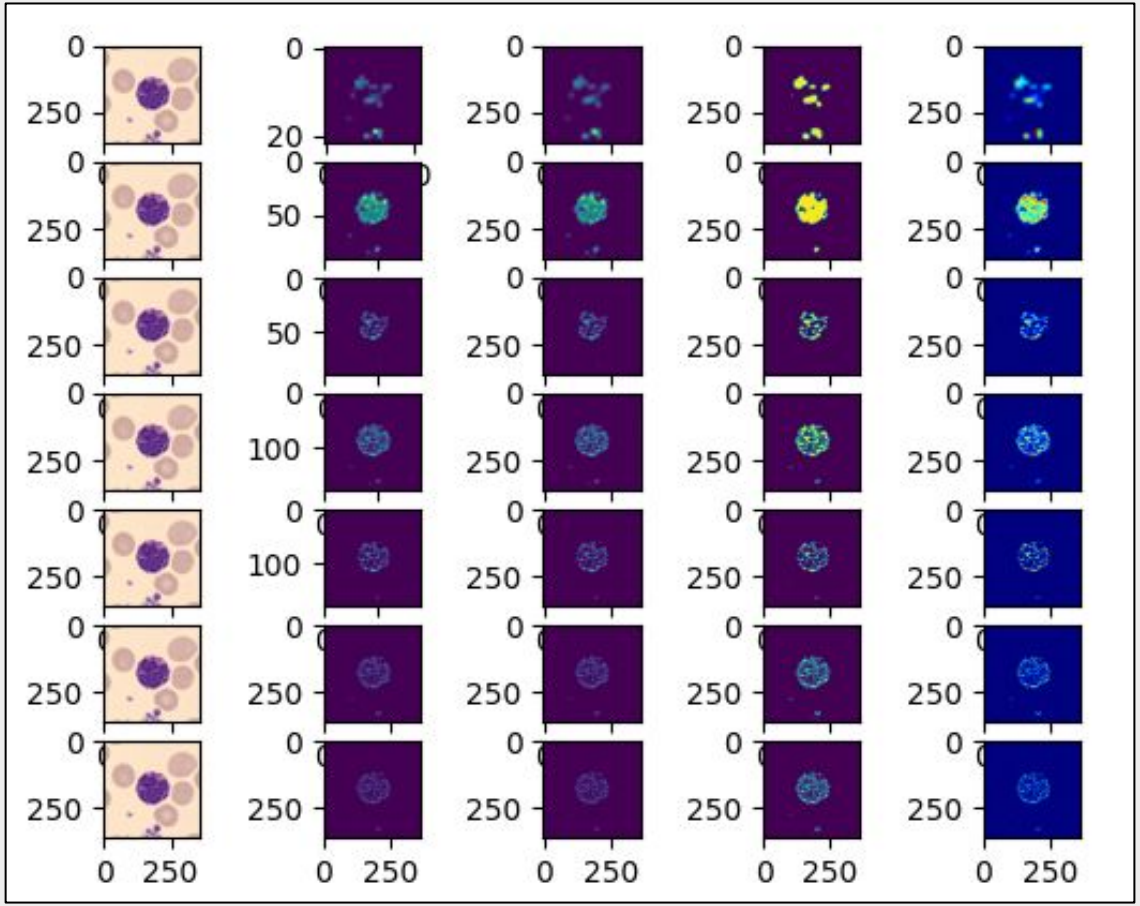
Examples of GRAD-CAM (Picture Courtesy: Dhruv Batra)

Masking using Deep Learning based methods via **GRAD CAM**

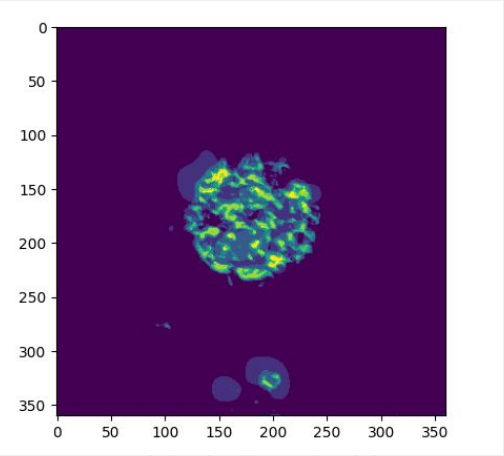
- We have used **GRAD CAM** to get confidence **segmentation masks** from each layer of the proposed Network.
- The proposed network for deriving the segmentation mask is shown in right.
- Firstly we get all the **confidence score** from each of the **convolutional layers**.
- We add each of the masks.
- We **threshold** the **masks** by the maximum value obtained from the masks.
- We apply **dilation** via **box kernel** of size 5x5



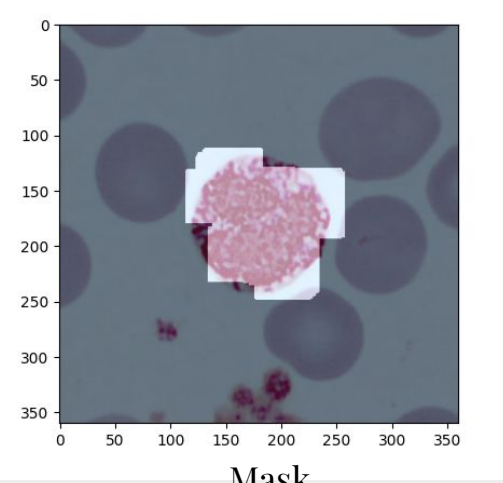
Masking using Deep Learning based methods via **GRAD CAM**



Images from each layers

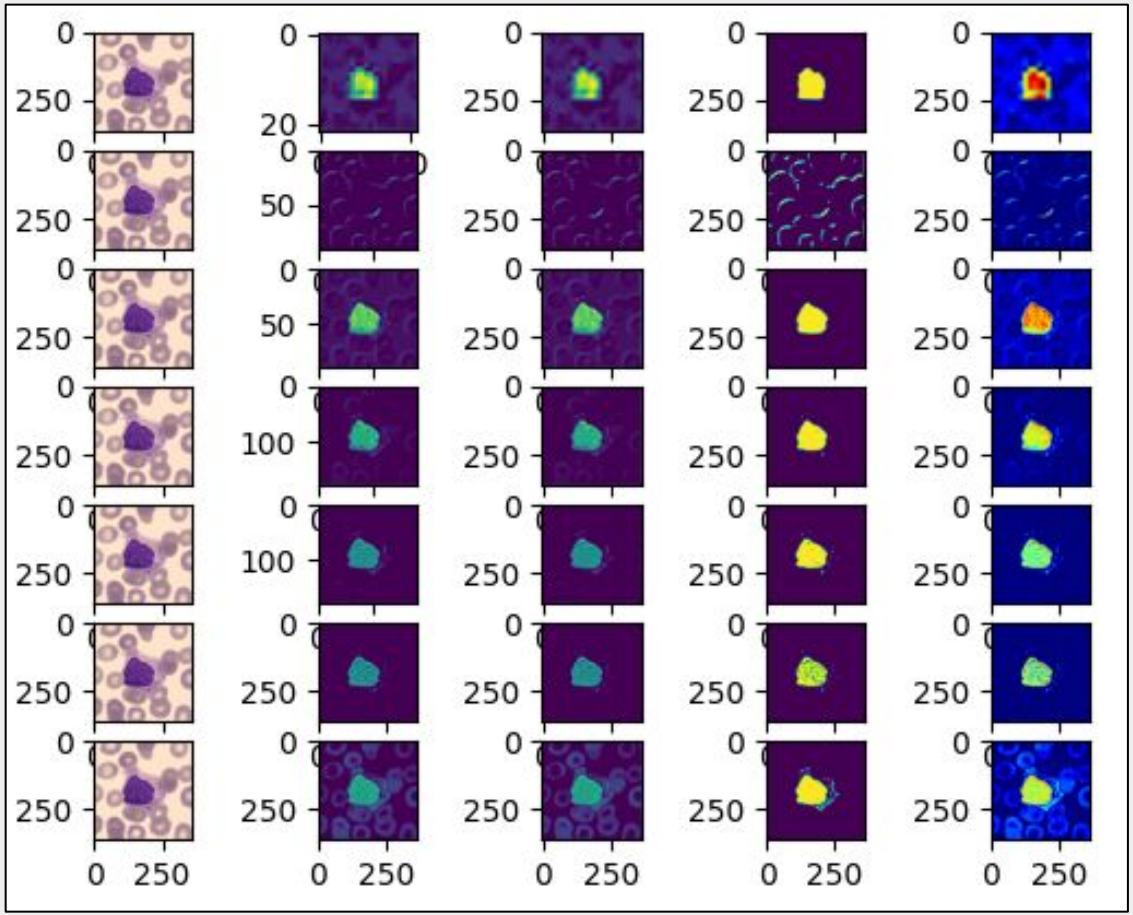


Added Threshold

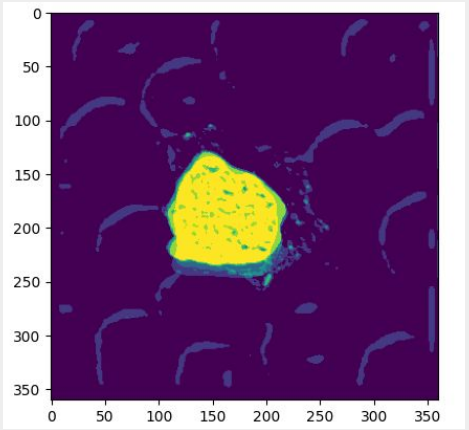


Mask

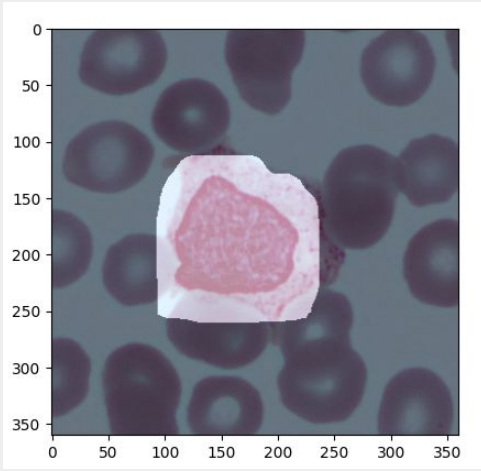
Masking using Deep Learning based methods via **GRAD CAM**



Images from each layers



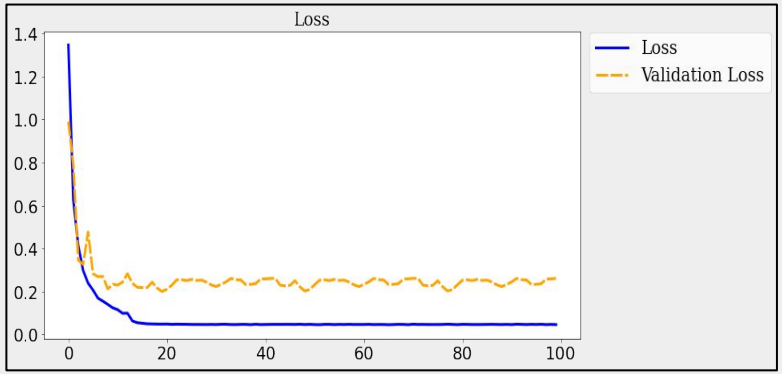
Added Threshold

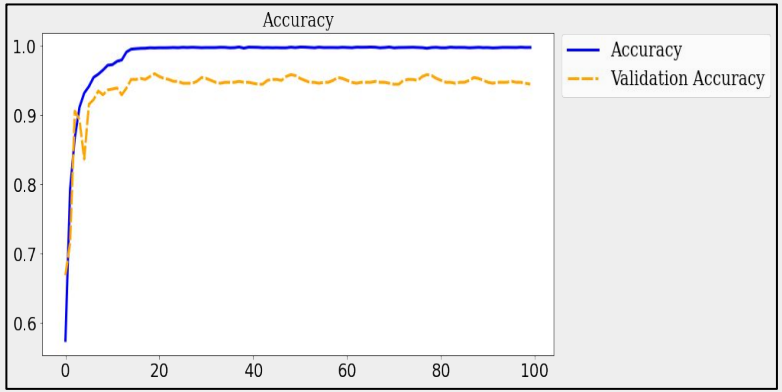


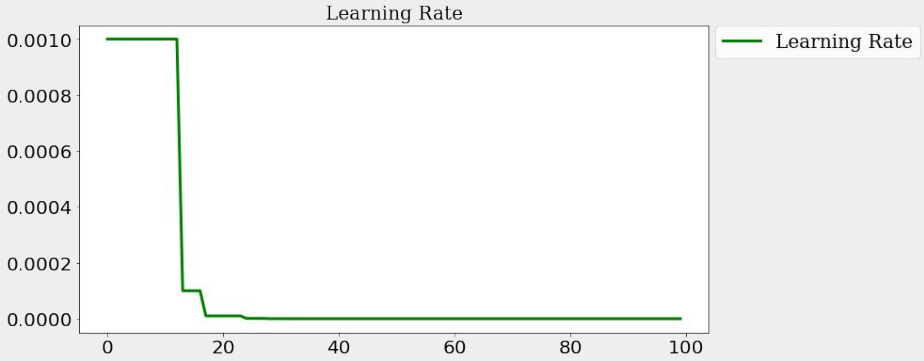
Mask

Proposed Model - Training Metrics

- Training for about 100 epochs we get a test loss of 0.3405 - and an accuracy of: 0.9440
- The plots of training and validation - accuracy and losses are shown below.
- We can see that the validation accuracy is a bit lower, this is probably due to the class imbalance.
- Change of learning rate is also shown below





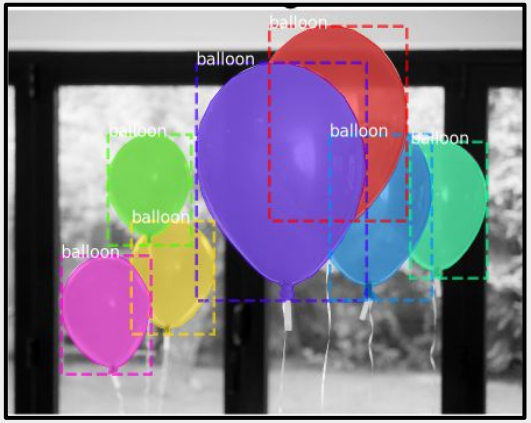


Segmentation techniques used

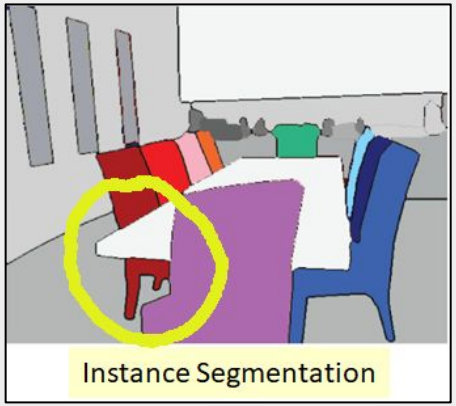
1. Watershed Algorithm
2. DoH, DoG and LoG
3. Grad CAM based Novel method
4. **Mask RCNN**

What is **Instance Segmentation** ?

- Recognizing and understanding what's in the image in **pixel level**.
- Assigning **different colours** to **different instance of the same object**.
- Here, different instances of balloons are assigned different colours.
- This is a hard task, since objects might be **occluded** and it is necessary to find different parts of the same object
- Example: Chair is occluded in the picture, but instance segmentation correctly recognizes it!



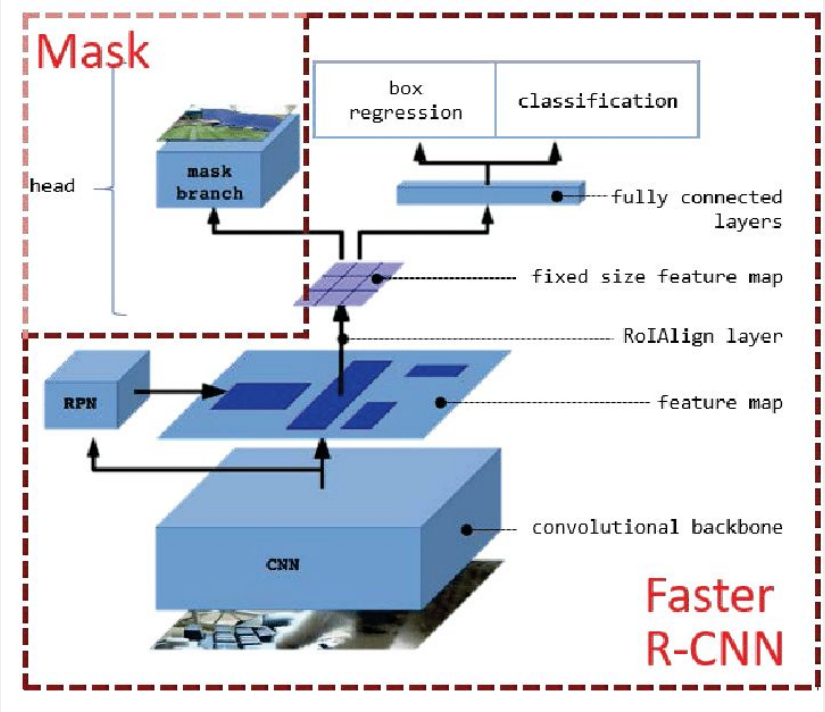
Instances of Baloons Segmented (Picture Courtesy: Medium)
https://miro.medium.com/max/886/1*I5_PSpXNmPDp5K9zvUZLpg.png



Instances of Chairs with Occlusion (Picture Courtesy: Medium)
https://miro.medium.com/max/2066/1*JpMn7MNESnA3lYmpdIcpUw.png

Mask RCNN in brief

- Gives object **bounding boxes**, **classes** and **masks** (**instance segmentation**) for an image.
- Convolutional backbone - **Feature Pyramid Network** (FPN) for preserving features at different scales - uses a Faster R-CNN backbone.
- **Region Proposal Networks** (RPN) contains bounding boxes, known as anchors, which helps to detect objects faster.
- Uses **ROIAlign** to align the features at different scales using **Bilinear Interpolation**, which helps to **remove location misalignment** caused due to **ROI pooling**.



Mask RCNN (Picture Courtesy: Research Gate)
https://www.researchgate.net/figure/The-structure-of-the-Mask-R-CNN-architecture_fig2_337795870

Mask RCNN in brief - Results on various datasets

- Better and accurate feature maps can be obtained as shown in the figure.

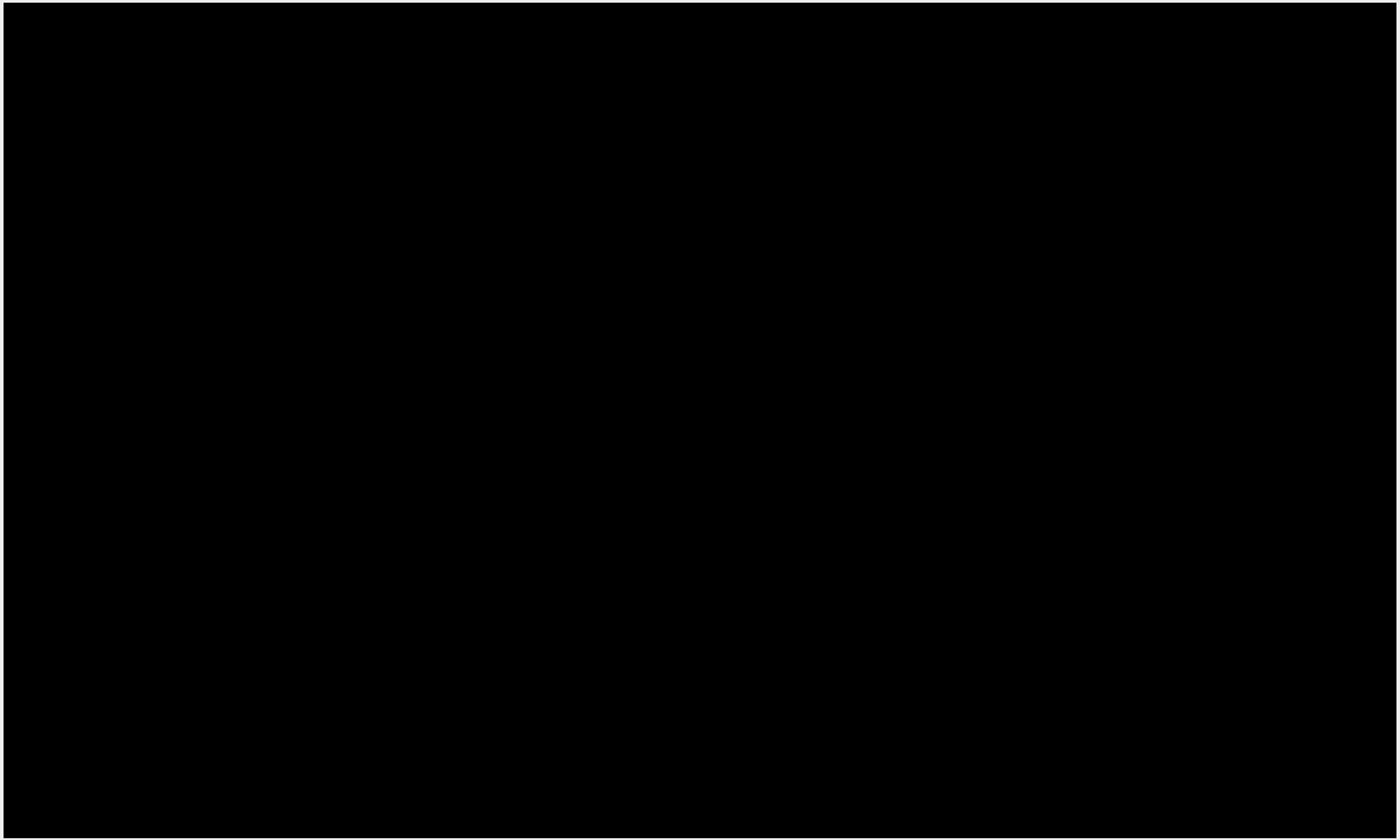
Mask RCNN applied on COCO test set
(Reproduced with permission from Kaiming He)

Mask RCNN applied on CityScape dataset
(Reproduced with permission from Kaiming He)

Mask RCNN

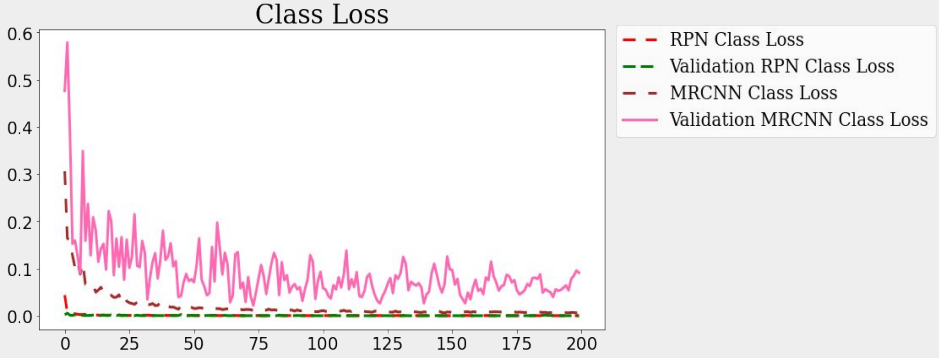
- We have trained the model for 400 epochs, 200 for network heads and 200 for full network.
- 0.7 was set as minimum confidence level, and 128 ROIs were used.
- A script was written which could generate a JSON file containing all the annotations of images in a given folder
- This script was extended for helping in creation of polygons for unseen images, which could also imported in CVAT, hence automating the process of annotation.
- The demonstration of the network is shown in the next slide, where it is given a set of test images for which it generates a set of mask in JSON format

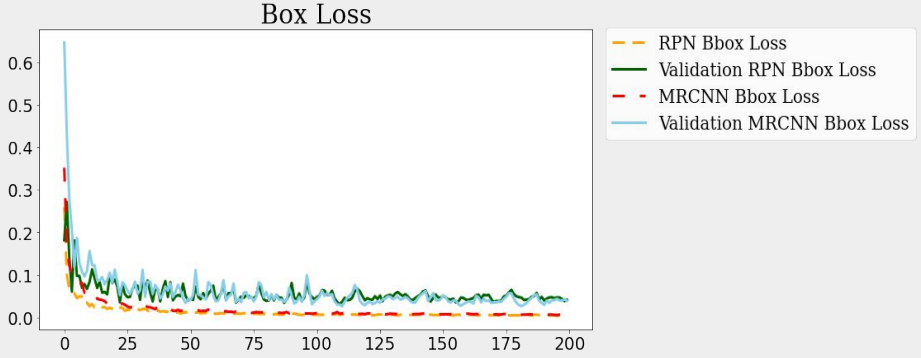
Mask RCNN -Working

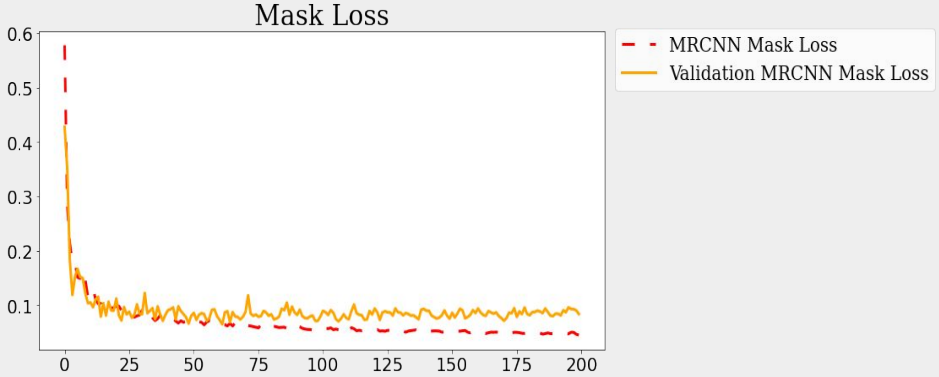


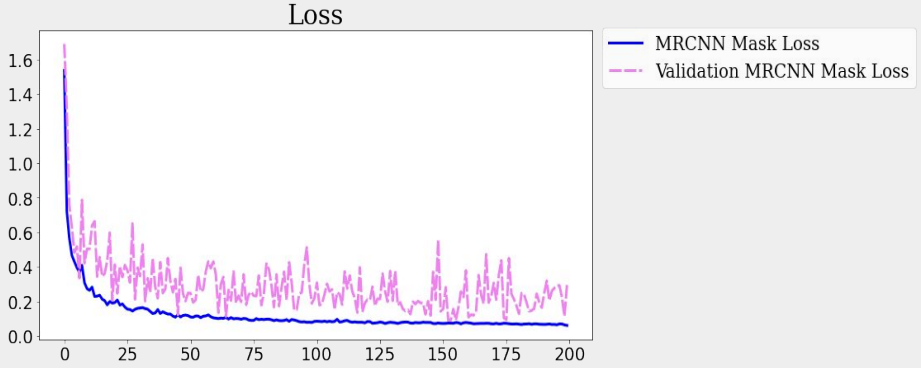
MASK RCNN - losses (Network Heads)

- Four types of losses have been recorded, one for network heads and another for training the whole network



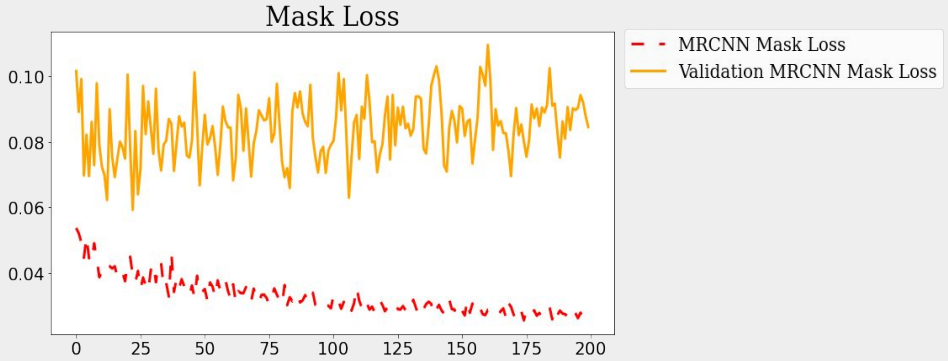


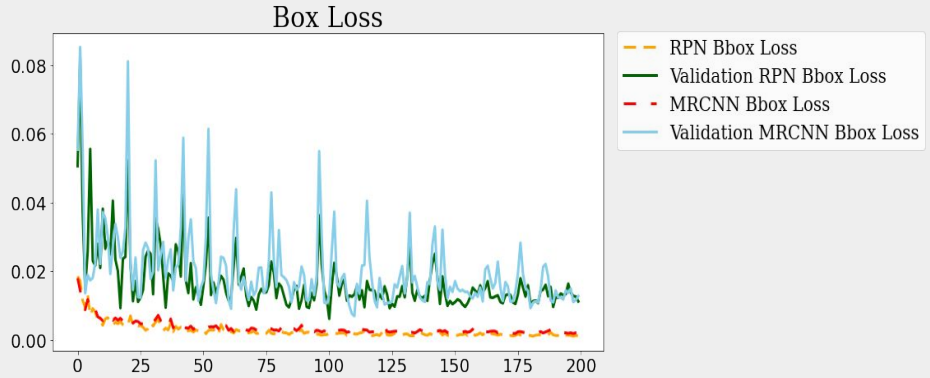


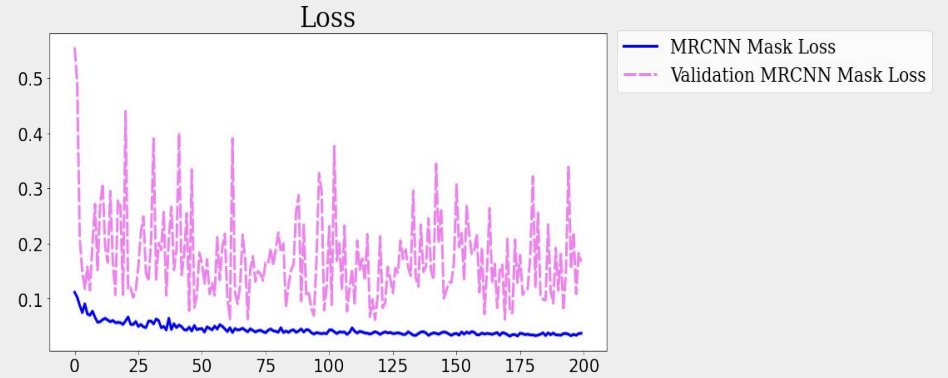


MASK RCNN – losses (Full Network)

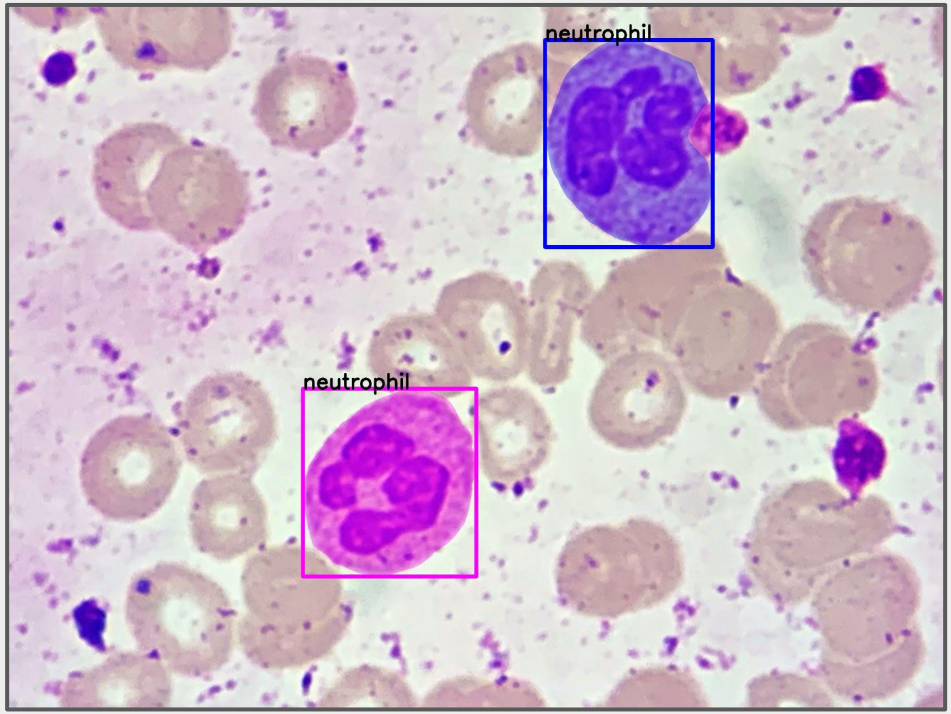
- Total loss can be written as: $$L_{total} = L_{cls} + L_{box} + L_{mask}$$



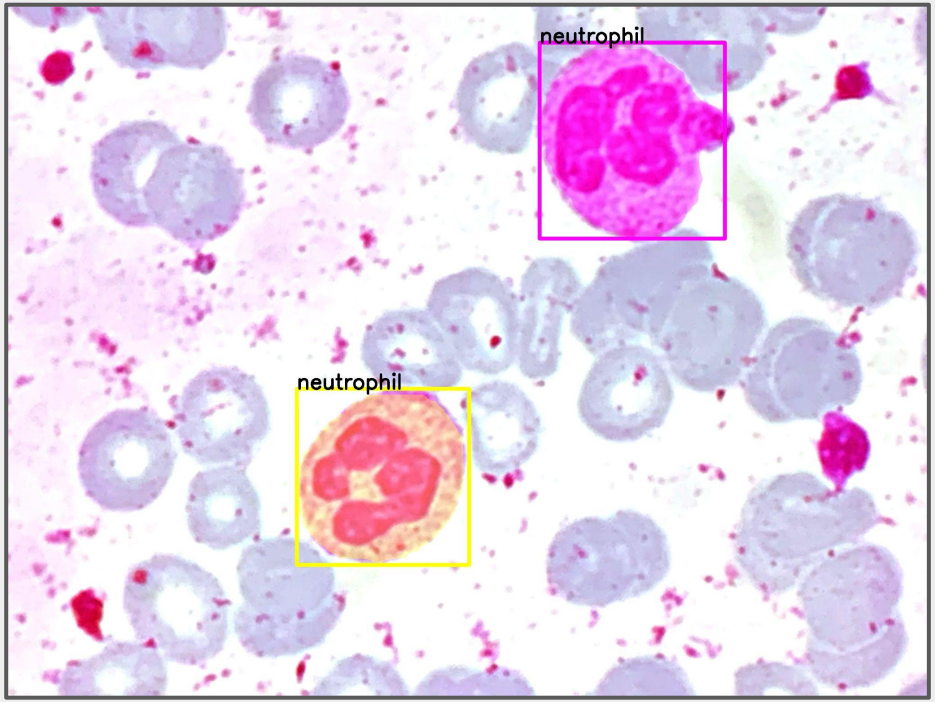




MASK RCNN Results - On Smear Slides Test Dataset (1)

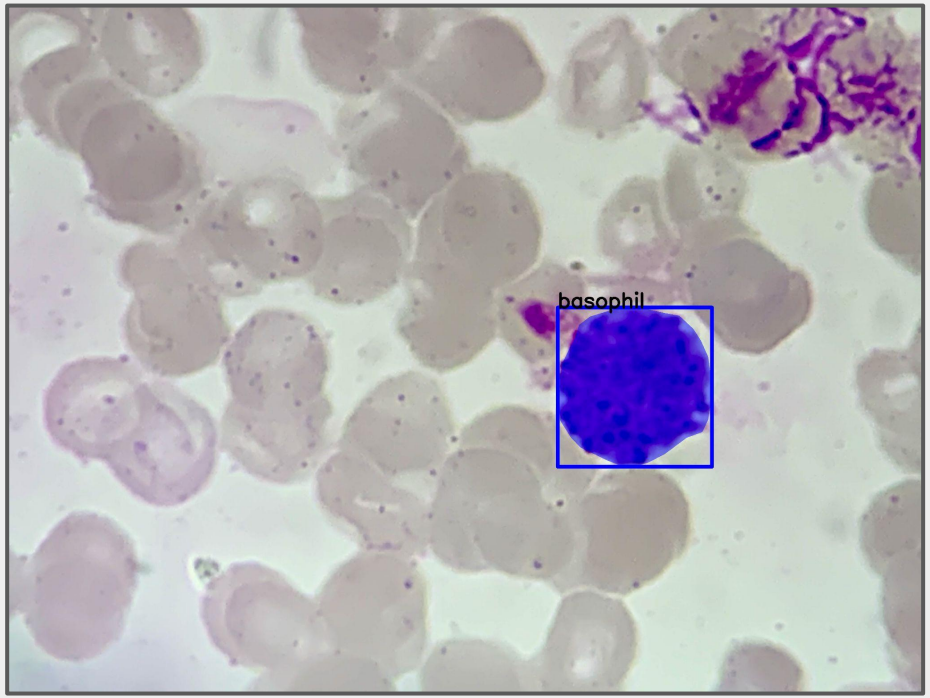


Ground Truth Image

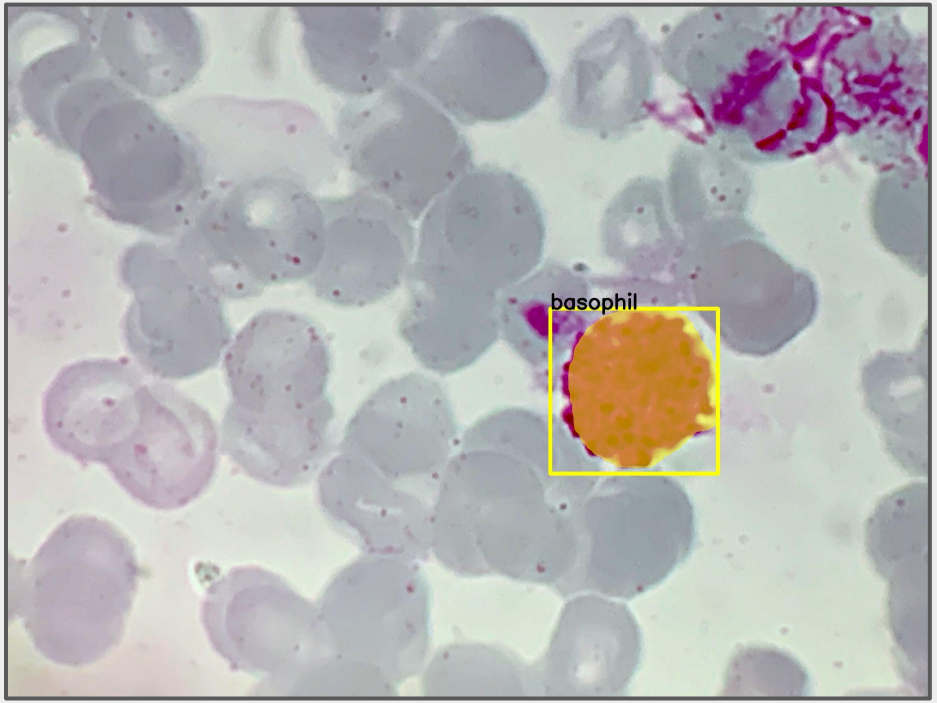


Mask RCNN Detected Image

MASK RCNN Results - On Smear Slides Test Dataset (2)

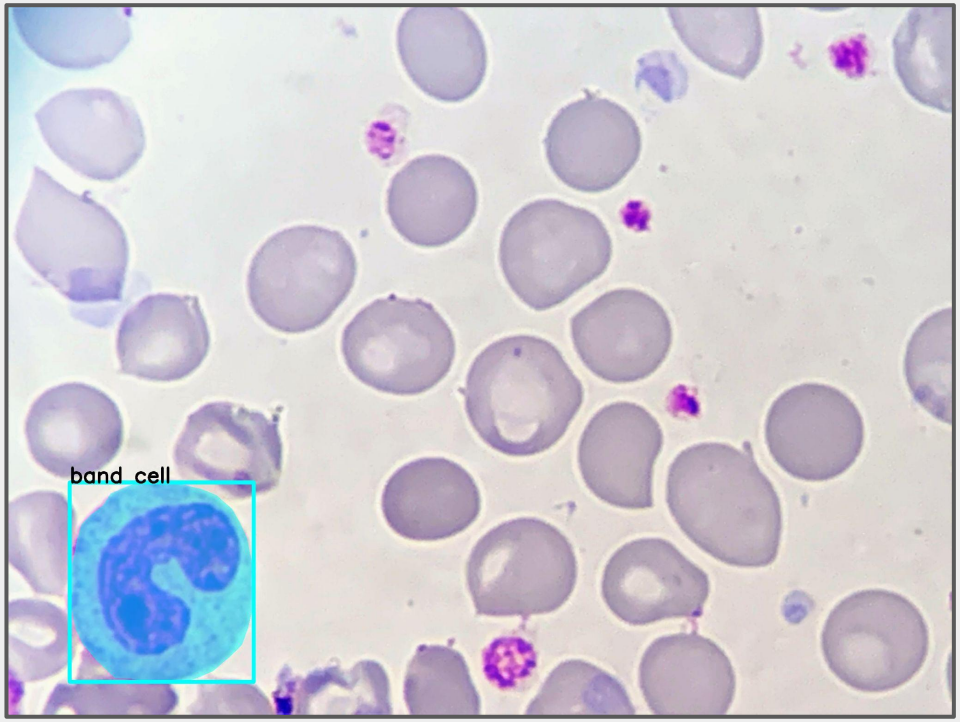


Ground Truth Image

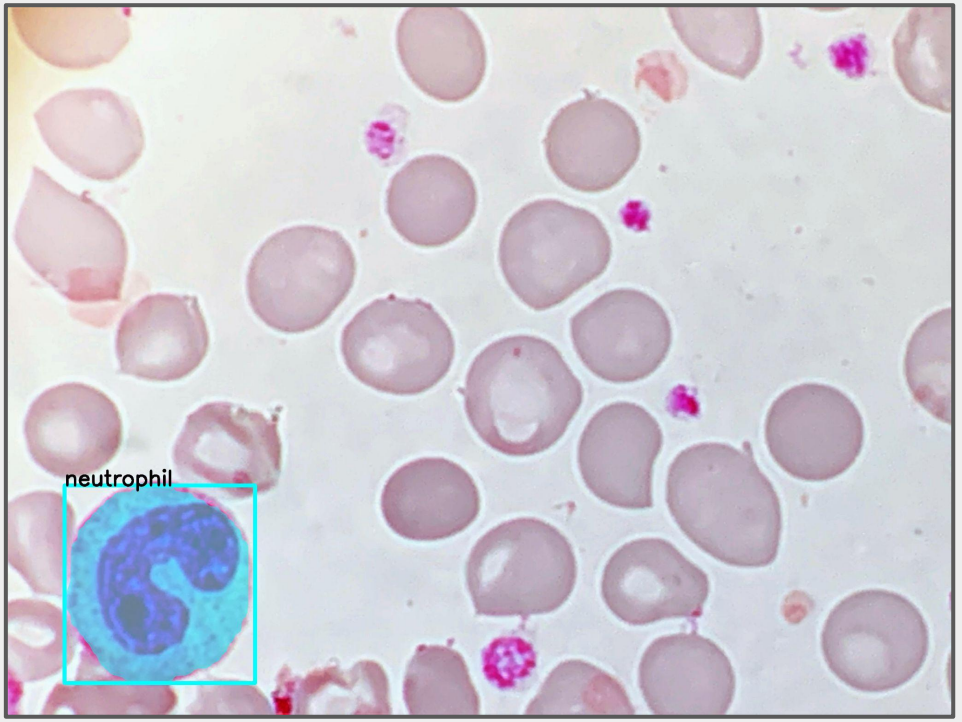


Mask RCNN Detected Image

MASK RCNN Results - On Smear Slides Test Dataset (3)



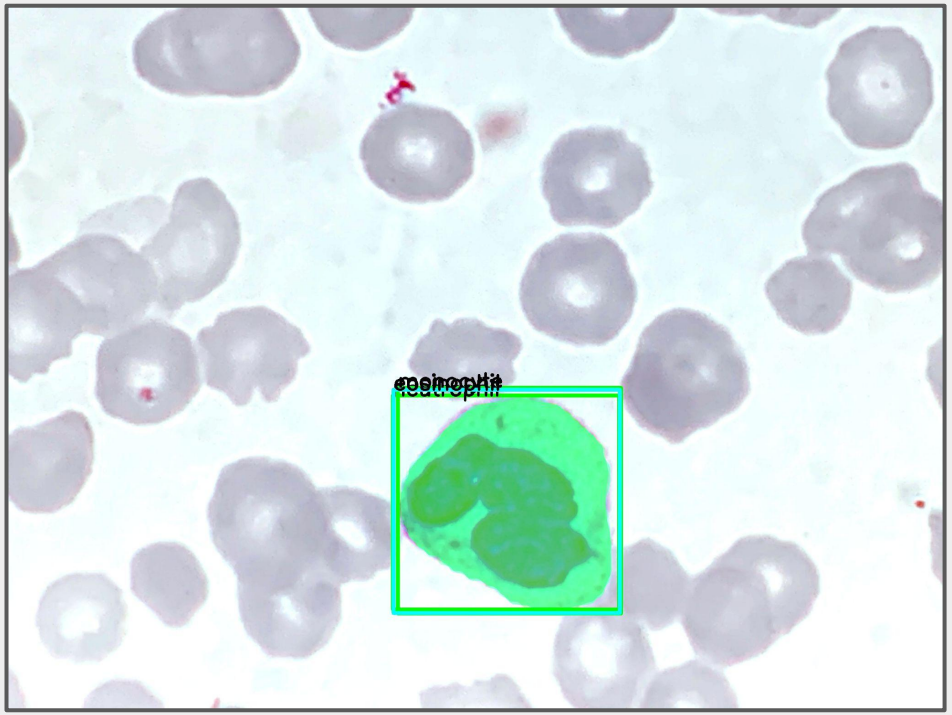
Ground Truth Image



Mask RCNN Detected Image

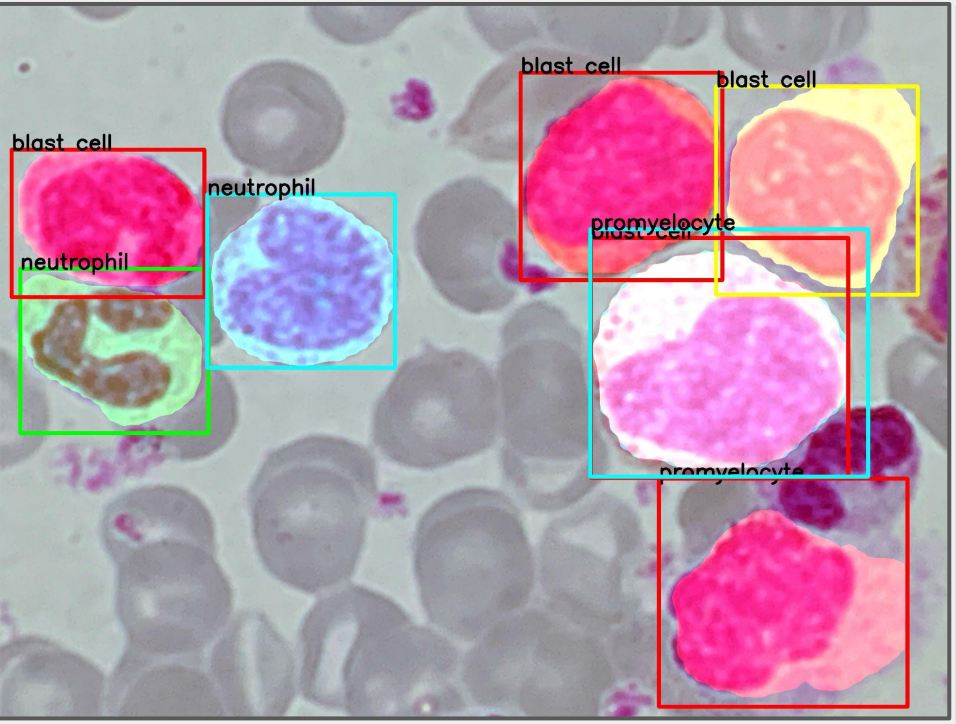
MASK RCNN Results - On Smear Slides Test Dataset (4)

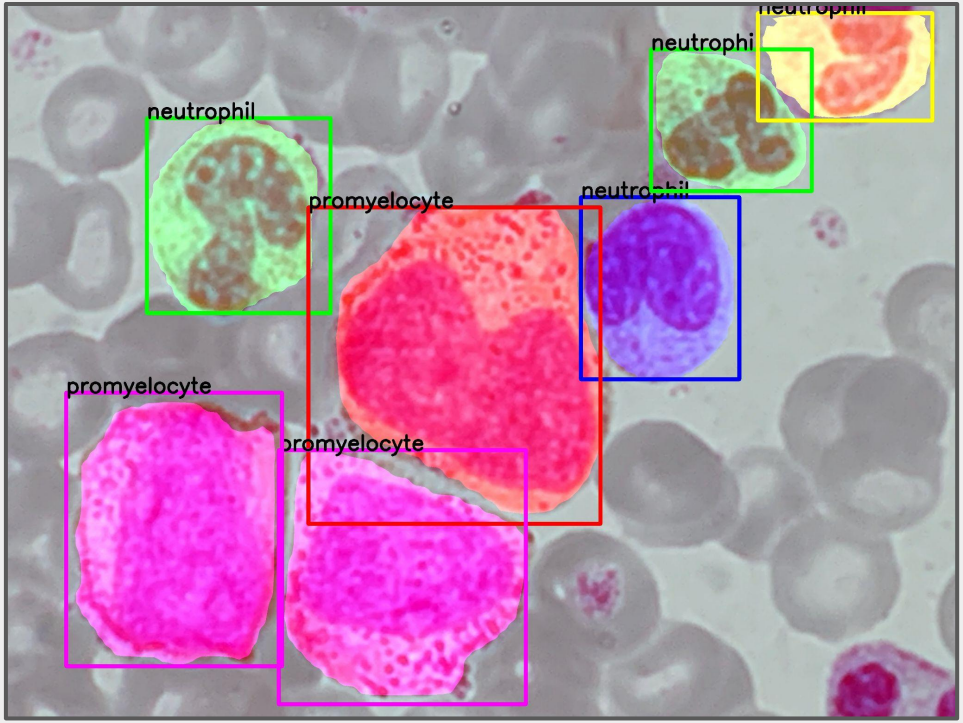
Ground Truth Image



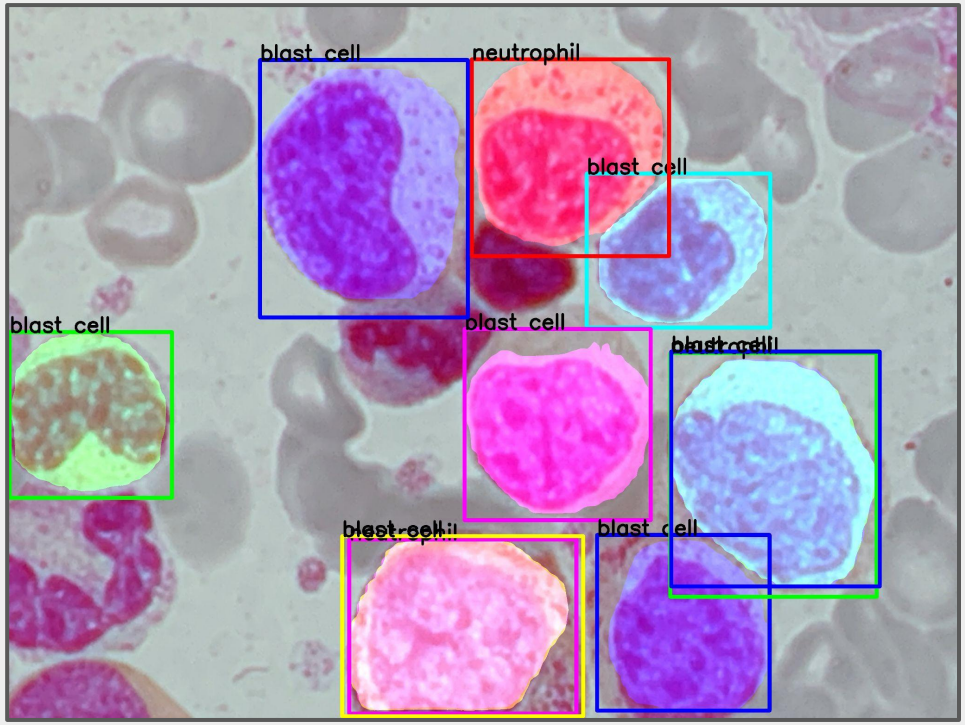
Mask RCNN Detected Image

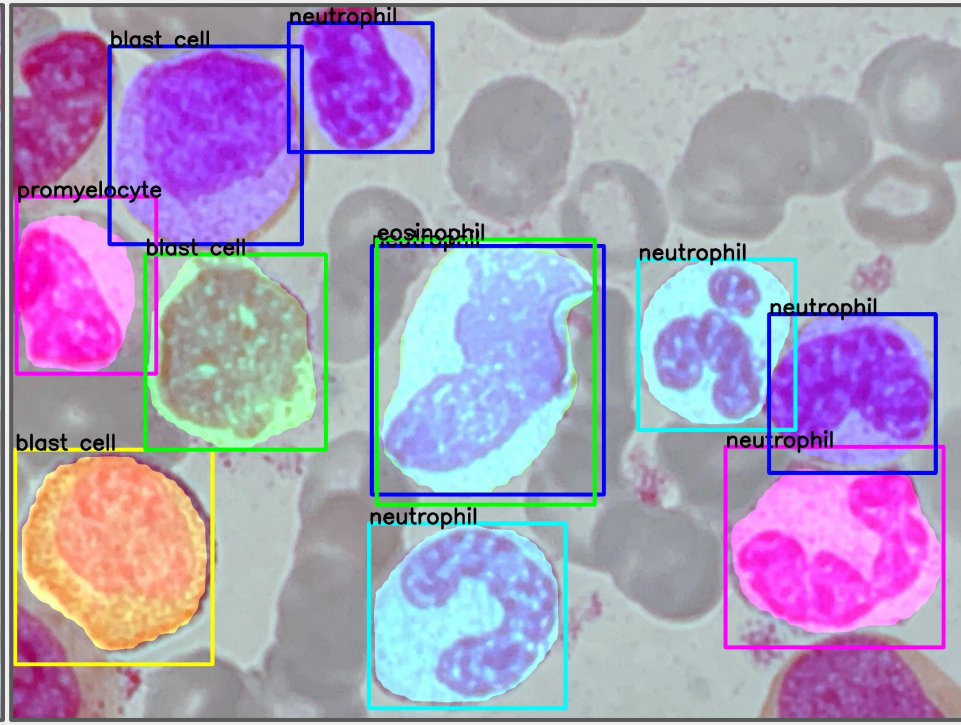
MASK RCNN Results - On Smear Slides Mixed (1)



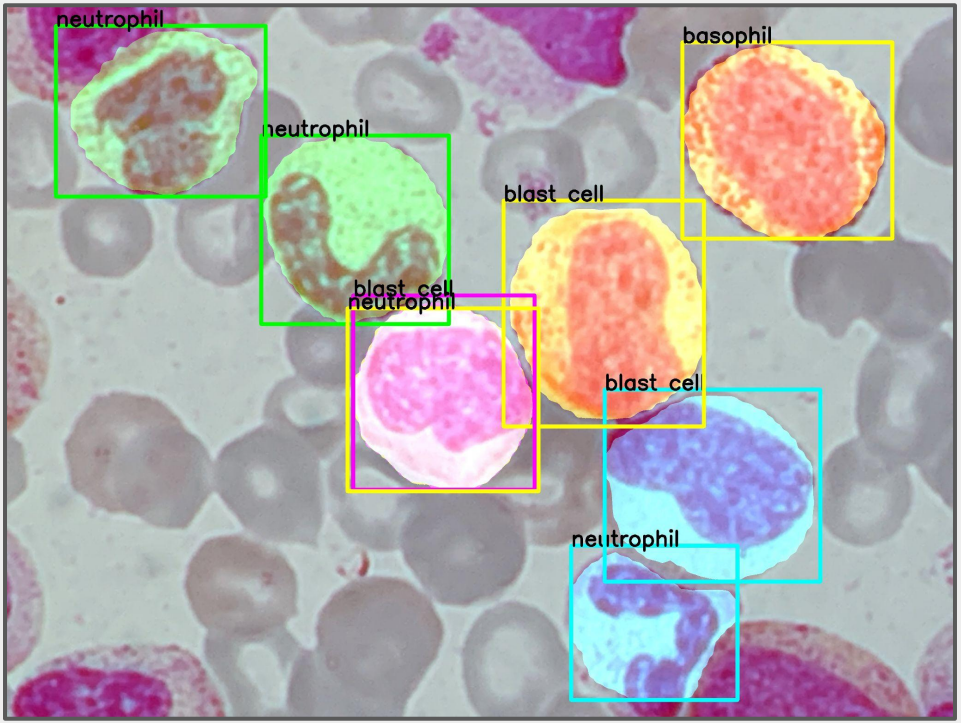


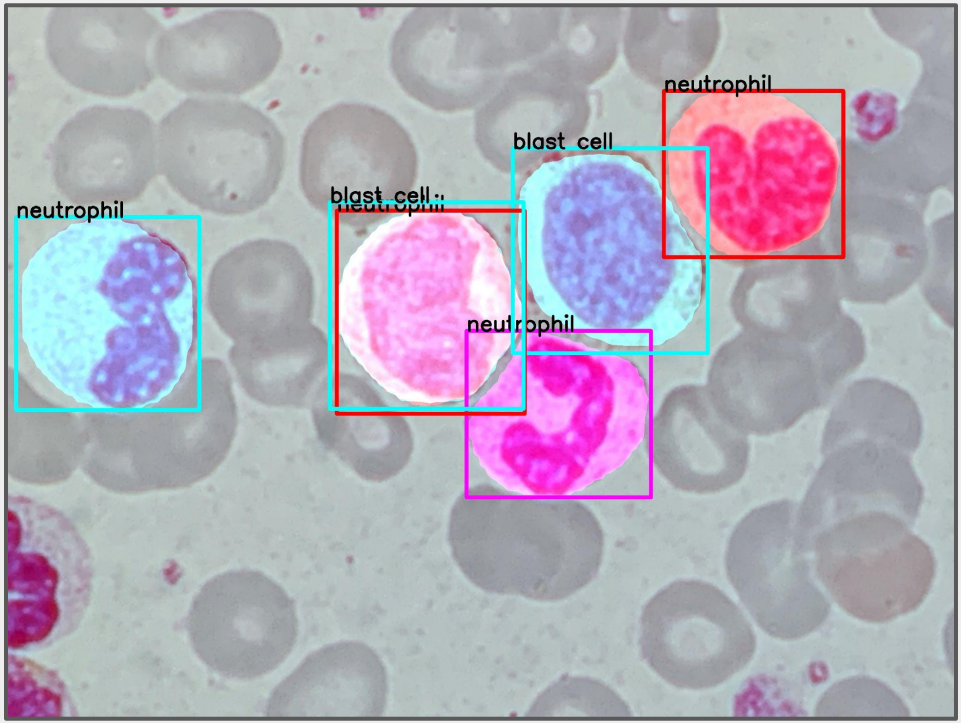
MASK RCNN Results - On Smear Slides Mixed (2)



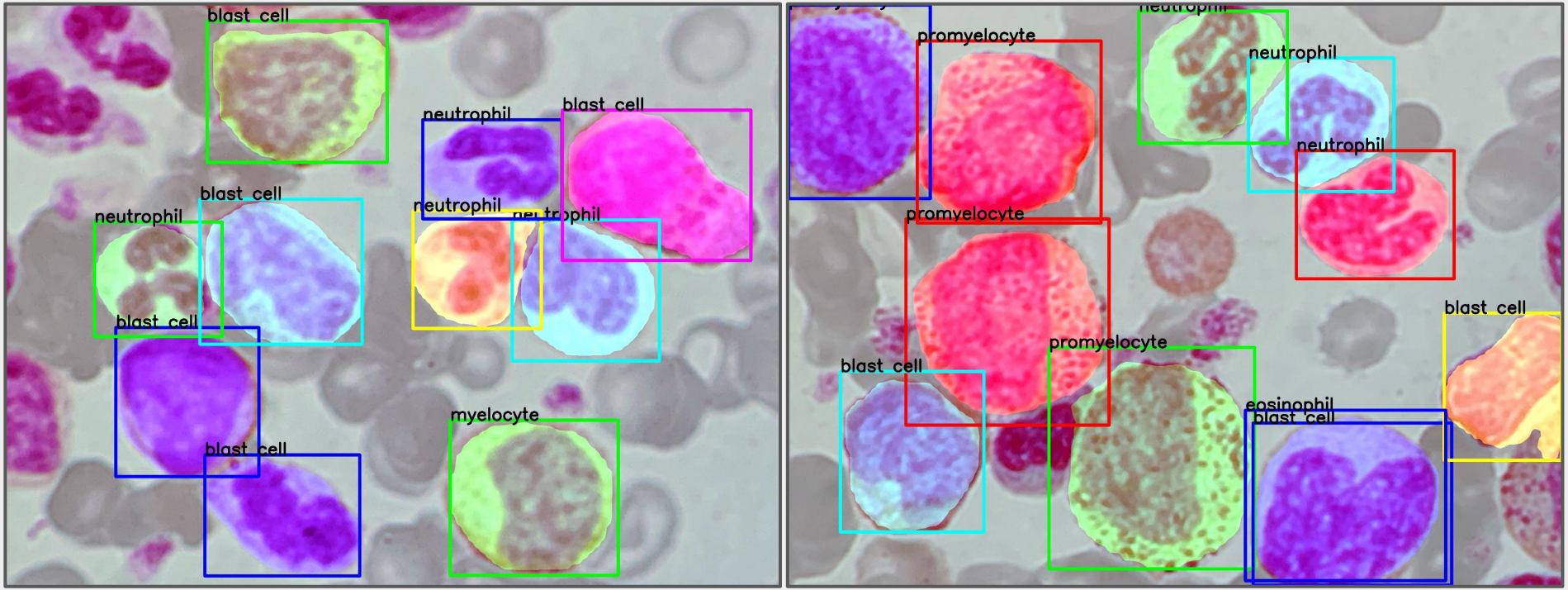


MASK RCNN Results - On Smear Slides Mixed (3)





MASK RCNN Results - On Smear Slides Mixed (4)

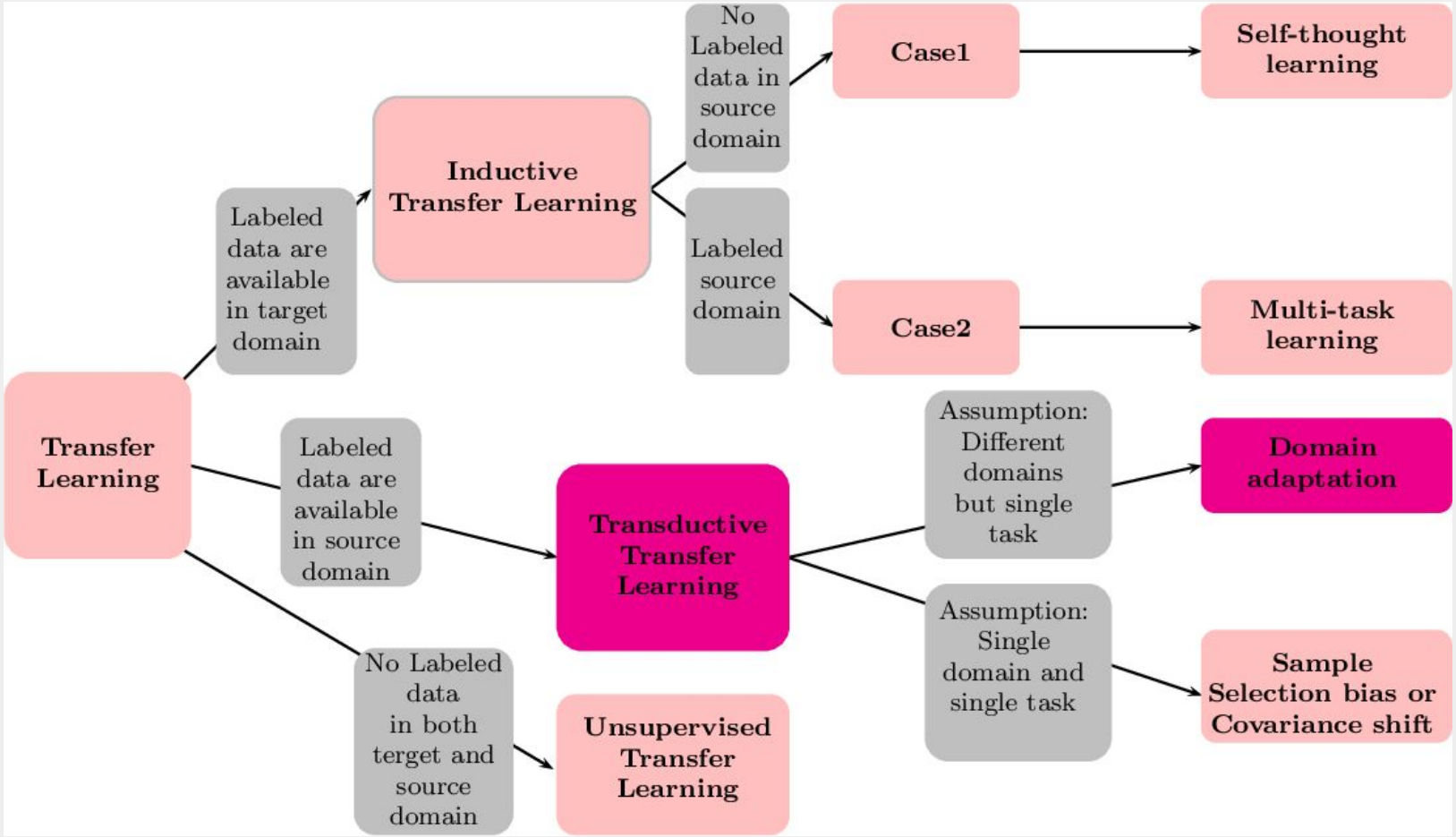


Mask RCNN - Generated JSON format

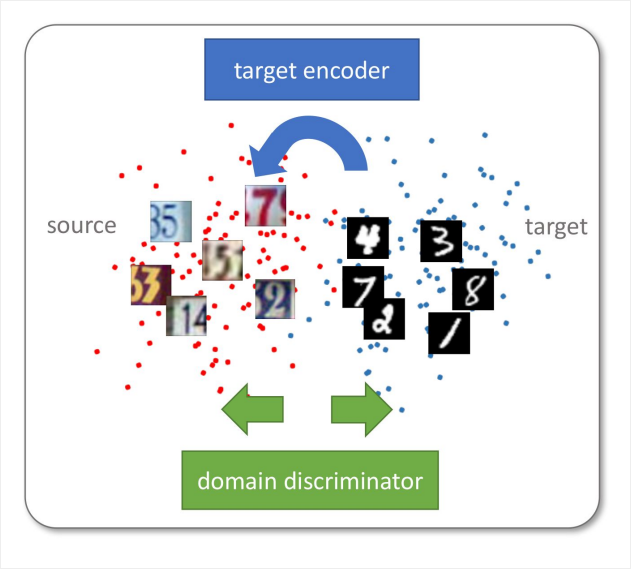

```
{"data":
        {"0":
                {"filename":"IMG_4302.jpg",  "height":3024, "width":4032,
                "masks":
                [ {"Class_name":"neutrophil",  "score":0.9632735252,
                        "bounding_box":{ "x1":22, "x2":740, "y1":436, "y2":1169},
                        "vertices":[ [397.0,1142.5], ... ,  [396.0,1141.5],[397.0,1142.5]]},
                {
                        "class_name":"neutrophil",
                        "score":0.9132735252,
                        "Bounding_box": {"x1":223, "x2":940, "y1":936, "y2":1769},
                "vertices":[[223.6,940.5], ... ,[224.3,939.4],223.6,940.5]]}
                ]},
        "1":
                {
                ...
                }       ...
        }
}
```

Domain Adaptation

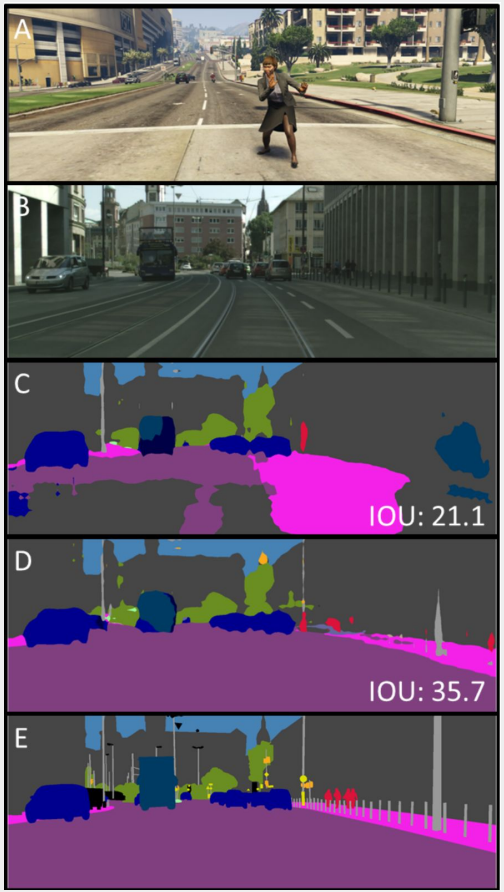
Domain Adaptation - As a part of Transfer Learning



Domain Adaptation - Related Works



MNIST to **SVHN** and vice versa [11]



GTA V to **CITYSCAPES [11]**

Domain Adaptation - Motivation

- Since we have very less data, we will use a different dataset of similar kind to enhance the detection of related classes for the Smear Slides dataset.
- We are using Smear Slides Cropped dataset for one domain, and PBC Cropped for another domain.
- For performing Domain Adaptation, we will select PBC Cropped as source and Smear Slides as destination once, and vice versa.
- The first method will enhance the classification of PBC Cropped dataset, whereas the second one will make the 8-class classification for Smear dataset superior.
- First, let us look at the formulation for performing domain adaptation.

Domain Adaptation - Formulation

- Let there be two complex unknown distribution, Source ***S(x,y)*** and Target ***T(x,y)*** where, **x** ∈ **X,** from the input space and **y** ∈ **Y** from label space.
- Distribution **S** is shifted from target **T** by some domain shift.
- An extra parameter is introduced, i.e., the domain **d** :

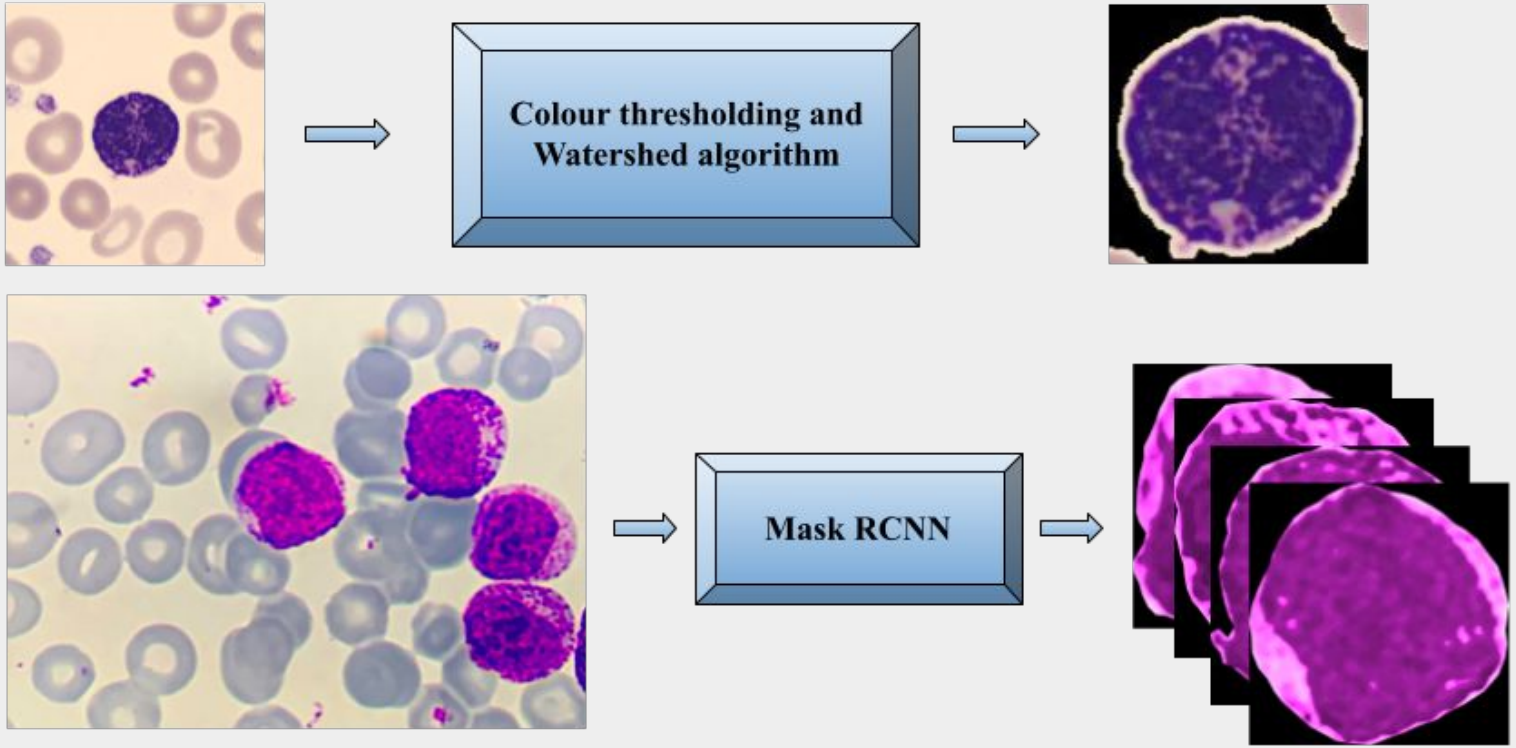
$$d_i = \begin{cases} 0, \text{ if } x_i \text{ is in Source domain, } y_i \text{ is in Source Domain and are } \textbf{known}. \\ 1, \text{ if } x_i \text{ is in Target domain, and } y_i \text{ from target domain and are } \textbf{not known}. \end{cases}$$

- Task is to predict target domains input sample at test time.
- During training, we aim to minimize the label predictor's loss and the feature extractor is optimized, hence the empirical loss of the source domain samples are minimized.
- We are wanting to make the two features domain invariant and at the same time make the distribution,

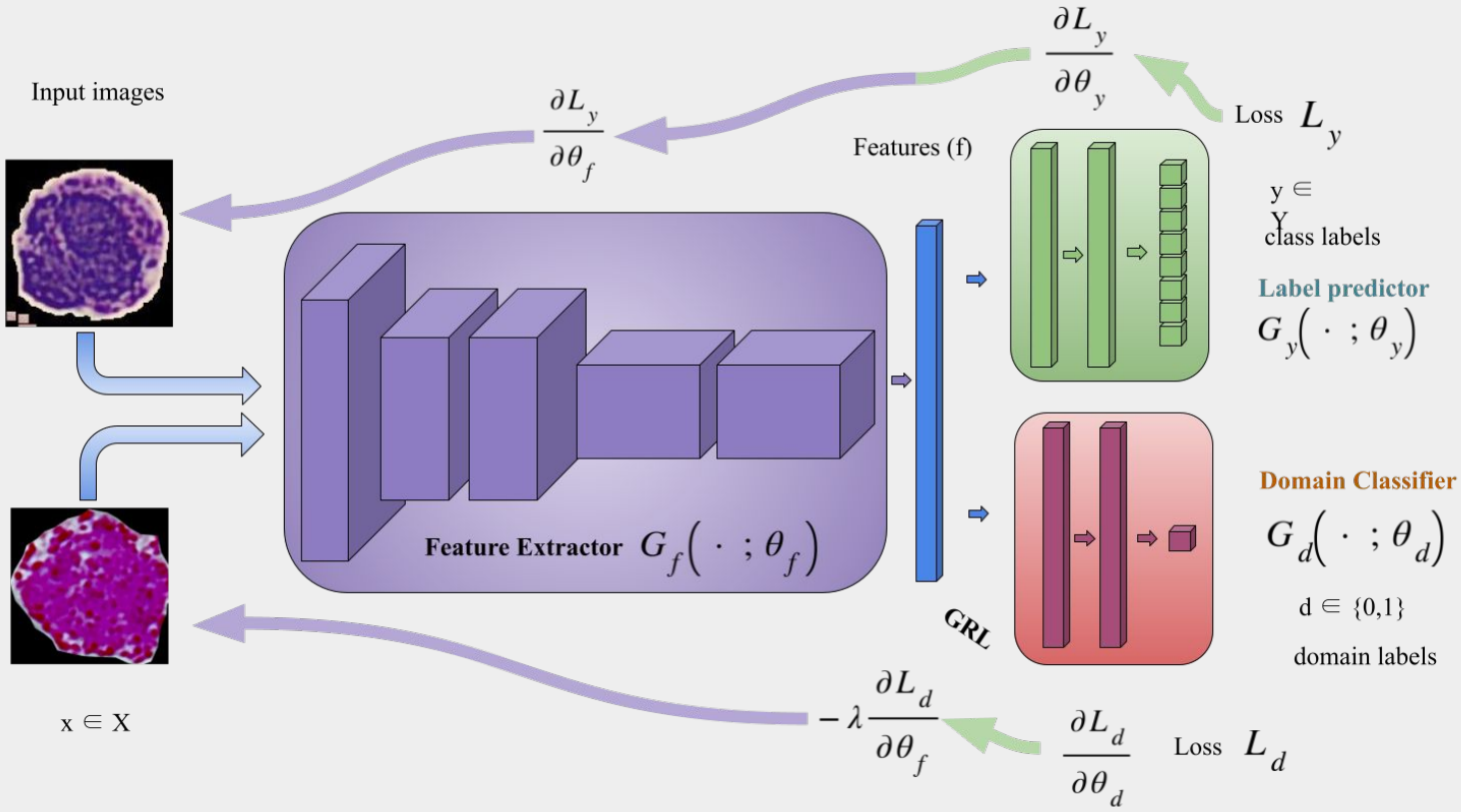
and distribution $S(f) = \left\{ G_f\left(x;\theta_f\right) \mid x \in S(x) \right\}$ as close $T(f) = \left\{ G_f\left(x;\theta_f\right) \mid x \in T(x) \right\}$

. to each other

Domain Adaptation - Our Approach (1. Detection)



Domain Adaptation Model - Unsupervised



Domain Adaptation - Formulation

- Measuring dissimilarity between distribution is nontrivial since f is high dimensional and the distribution themselves are constantly changing as learning progresses.
- Loss of domain classifier is maximized at the training time in order to obtain domain invariant features θ_f
- Hence, we minimize the loss for label predictor and maximize the loss for domain classifier.
- Hence the error can be formulated as:

$$E(\theta_f, \theta_y, \theta_d) = \sum_{i=1,2,..N,\ d_i=0} L_y\left(G_y\left(G_f(x_i;\theta_f);\theta_y\right), y_i\right) - \lambda \sum_{i=1,2,..N} L_d\left(G_d\left(G_f(x_i;\theta_f);\theta_d\right), y_i\right)$$

$$= \sum_{i=1,2,..N,\ d_i=0} L_y{}^i(\theta_f - \theta_y) - \lambda \sum_{i=1,2,...N} L_d{}^i(\theta_f, \theta_d)$$

- The label predictor here $L_y(.,.)$ is a classifier hence the loss is multinomial.
- Similarly, the domain discriminator loss $L_d(.,.)$ is logistic.

Domain Adaptation - Formulation

- $L_y{}^i$ and $L_d{}^i$ are the corresponding loss function that are evaluated at the *ith* training example.
- We are seeking parameters $\theta_f, \theta_y, \widehat{\theta}_d$ that deliver a saddle point to the function

$$\left(\widehat{\theta}_f, \widehat{\theta}_y\right) = \arg\min_{\theta_f, \theta_y} E\left(\theta_f, \theta_y, \widehat{\theta}_d\right)$$

$$\widehat{\theta}_d = \arg\max_{\theta_d} E\left(\widehat{\theta}_f, \widehat{\theta}_y, \theta_d\right)$$

- We may use standard optimizer such as Stochastic Gradient Descent for optimising, where μ is the learning rate

$$\theta_f \longleftarrow \theta_f - \mu\left(\frac{\partial L_y{}^i}{\partial \theta_f} - \lambda\frac{\partial L_d{}^i}{\partial \theta_f}\right) \qquad \theta_y \longleftarrow \theta_y - \mu\frac{\partial L_y{}^i}{\partial \theta_y} \qquad \theta_d \longleftarrow \theta_d - \mu\frac{\partial L_d{}^i}{\partial \theta_d}$$

Domain Adaptation - Formulation

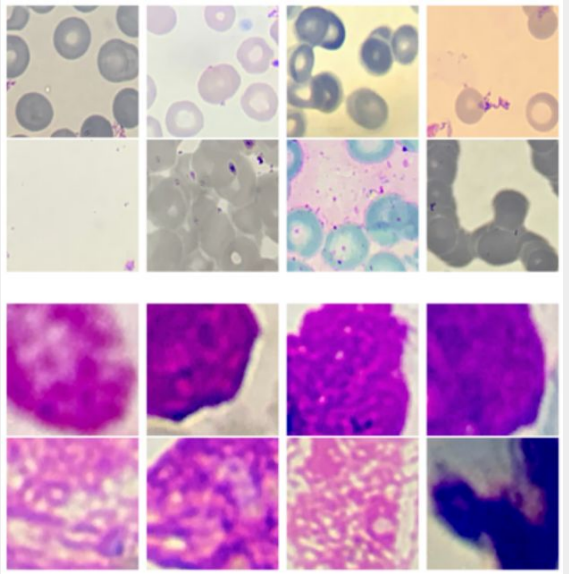
- The Equation $\theta_f \longleftarrow \theta_f - \mu\left(\frac{\partial {L_y}^i}{\partial \theta_f} - \lambda \frac{\partial {L_d}^i}{\partial \theta_f}\right)$ can be achieved by introducing a special Gradient Reversal Layer (**GRL**).
- Here λ is a meta parameter which is not updated by back-propagation.
- During forward propagation the **GRL** acts as an identity transform, but during back propagation the gradient from the subsequent layer is multiplied by - and passed to the preceding layer λ
- **GRL** is inserted between the domain classifier and the feature extractor.
- The total loss can be formulated as:

$$\widehat{E}\left(\theta_f, \theta_y, \theta_d\right) = \sum_{i=1,2,..N,\ d_i=0} L_y\left(G_y\left(G_f\left(x_i;\theta_f\right);\theta_y\right), y_i\right) + \sum_{i=1,2,..N} L_d\left(G_d\left(R_\lambda\left(G_f\left(x_i;\theta_f\right)\right);\theta_d\right), y_i\right)$$

- After learning, the label predictor $G_y\left(G_f\left(x;\theta_f\right);\theta_y\right)$ can be used to predict data from both the domains.

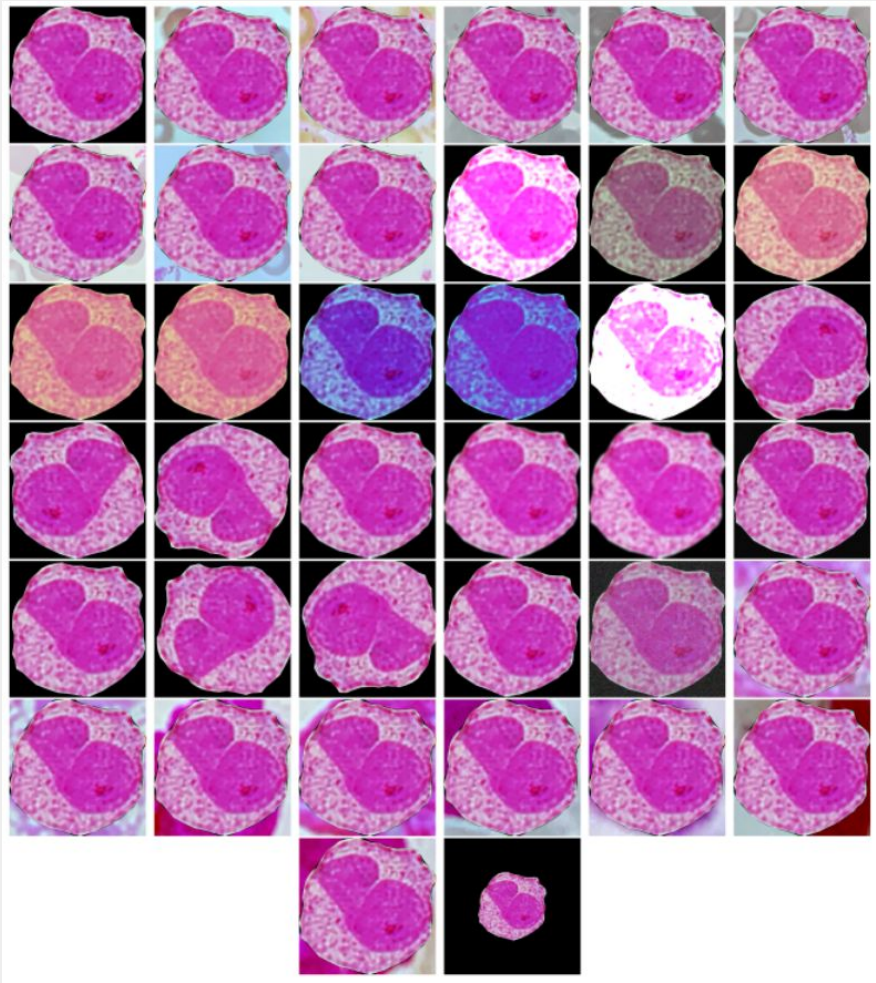
Domain Adaptation - Augmentation of Smear Slide Cropped Dataset

- It is clear that we need same amount of data from both the domains during training.
- For this we have selected the training dataset and performed various random augmentation on them.
- We have applied a series of transformations on the training dataset for Smear slides to make it similar size as that of PBC Cropped dataset.
- For augmentation, we have selected background of different textures from the Smear Slides data, to apply into the cropped dataset.
- This included normal background and stain background, for creating minor variations



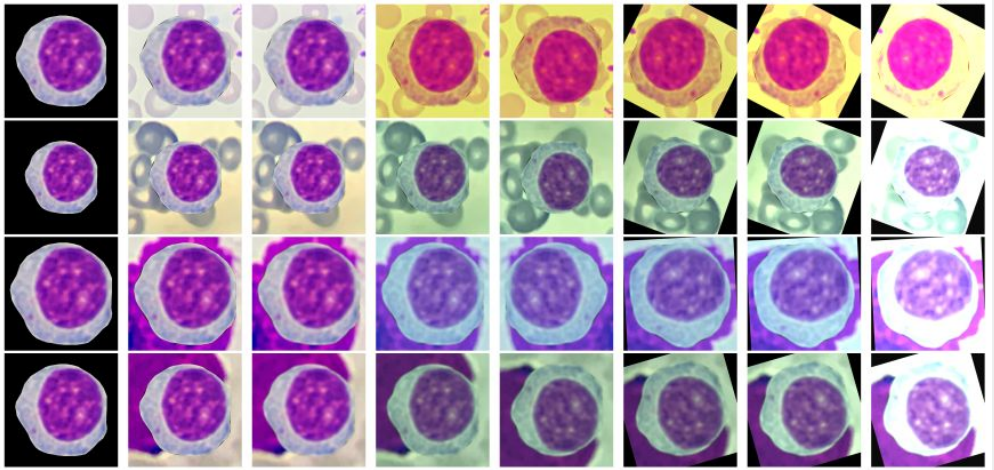
Data Augmentation - Smear Slides Cropped

- Augmentations applied on a single image of Monocyte cell.
- From top left: **(1)** Monocyte Cell, **(2-9)** Applying background on the cell, **(10)** Image with maximum brightness.
- **(11-16)** Applying different color variant to the image, **(17)** Image with maximum contrast, **(18-20)** different flips.
- **(21-23)** Gaussian blur repetitive with kernel sizes 5,11,17, **(24-27)** Rotating images, **(28)** Salt and Pepper noise,
- **(29-36)** Applying stain background, **(37)** Maximum zoom out of image. The transformations are applied to the original image, and in the augmentation module, these are applied recursively on a single image at random.



Data Augmentation

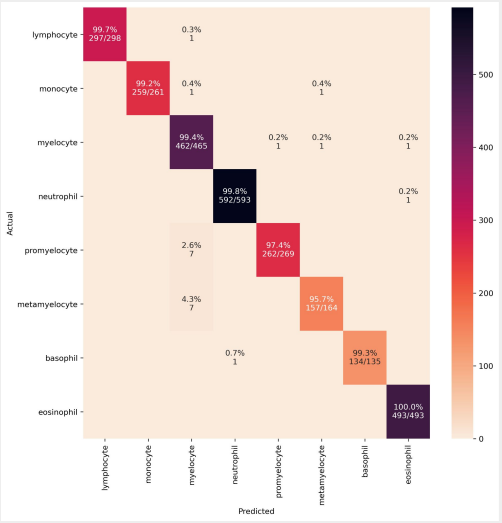
- A series of data augmentation might be shown in the figure (right).
- Medical images are generally taken under constraint conditions, for making the model more robust, we had to do the extensive data augmentation.

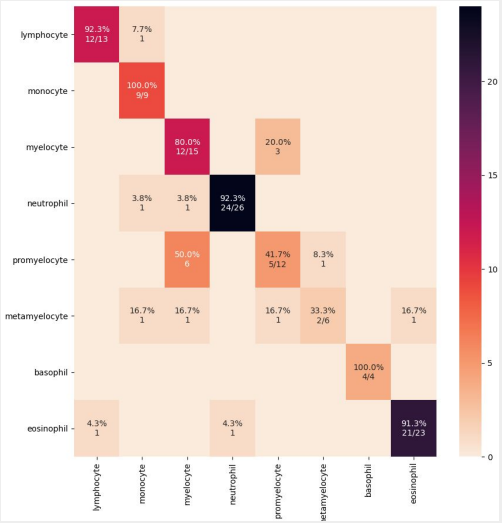


Data Augmentation - Results

- Domain Adaptation was performed from **PBC Cropped (Source) to Smear Slides Cropped dataset (Target)**.
- Confusion matrix for PBC Cropped (top) and Smear Slides Cropped (bottom) were recorded.

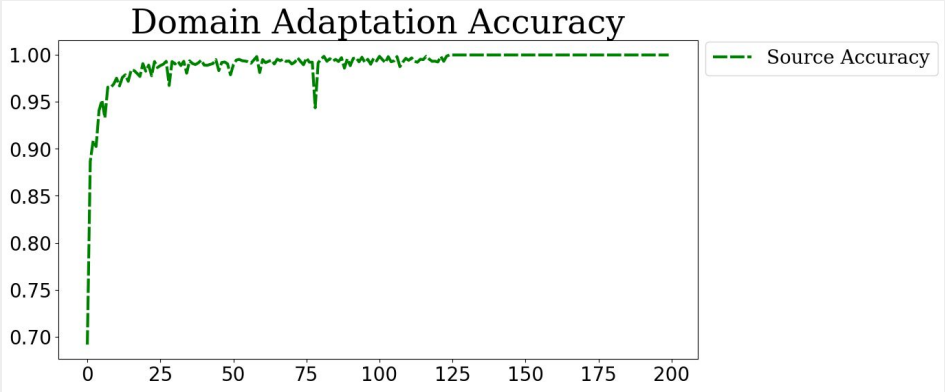
Remarks	Precision (in %)	Recall (in %)	F1-score (in %)
Domain Adaptation Source: PBC Cropped Target: Smear Slides 8 Cropped, Testing on Smear Slides Cropped	78.89	78.31	77.11
Domain Adaptation Source: PBC Cropped Target: Smear Slides 8 Cropped, Testing on PBC 8 Cropped	95.64	95.77	95.89

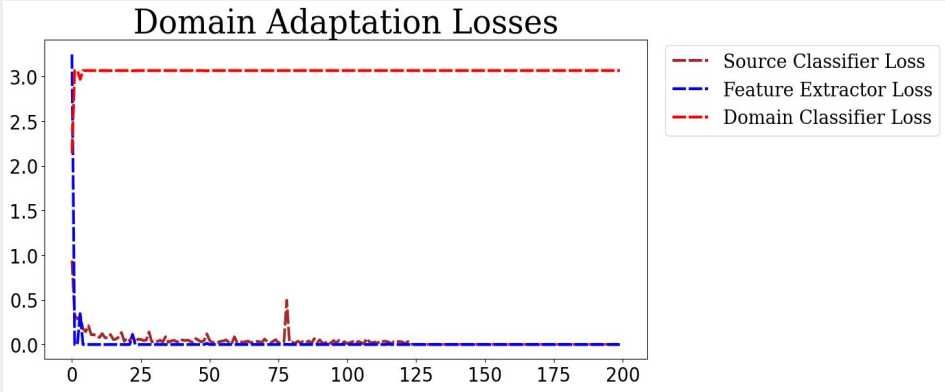




Data Augmentation - Results

- The variations of accuracy and losses were also recorded as shown below.

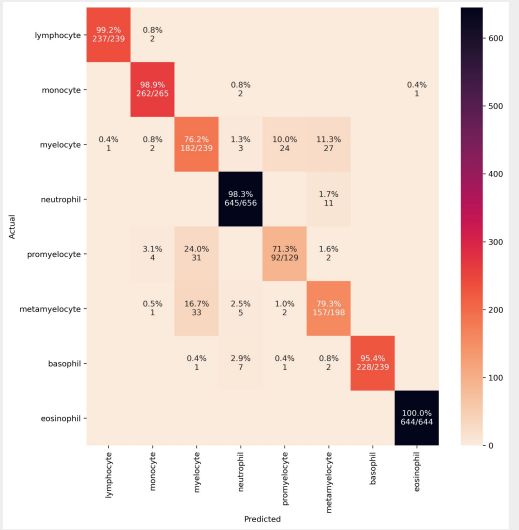


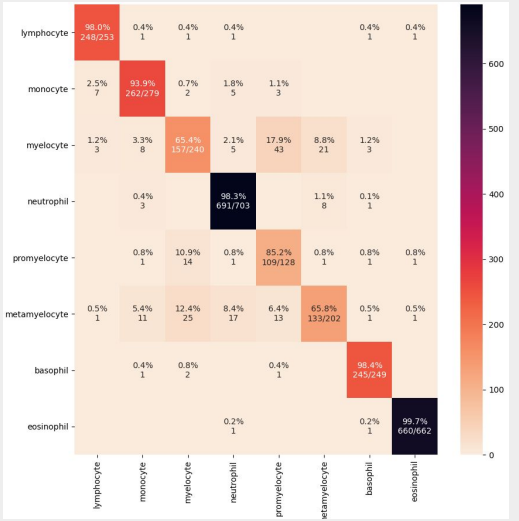


Data Augmentation - Results

- Domain Adaptation was performed from **Smear Slides Cropped dataset (Source) to PBC Cropped (Target)** .
- Confusion matrix for Smear Slides Cropped(top) and PBC Cropped (bottom) were recorded.

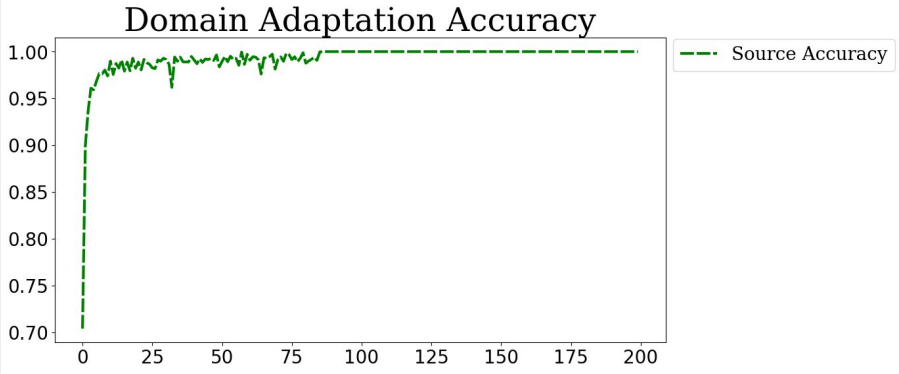
Remarks	Precision (in %)	Recall (in %)	F1-score (in %)
Domain Adaptation Source: Smear Slides Cropped Target: PBC 8 Cropped, Testing on Smear Slides 8 Cropped	86.71	85.67	84.89
Domain Adaptation Source: Smear Slides 8 Cropped Target: PBC 8 Cropped, Testing on PBC 8 Cropped	93.11	93.21	93.37

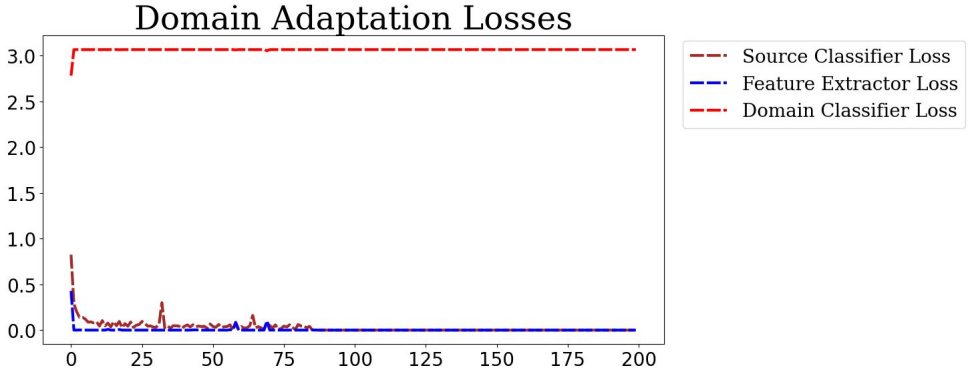




Data Augmentation - Results

- The variations of accuracy and losses were also recorded as shown below.





Results on the 8 classes

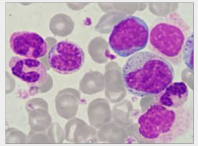
- Few experiments were conducted, and we found out that the maximum metrics were captured by Domain Adaptation modules.
- This happened because the model were learning different features which were robust to both the domains
- This is the maximum result we could generate via domain adaptation, obviously, more data would give better results on unseen data.

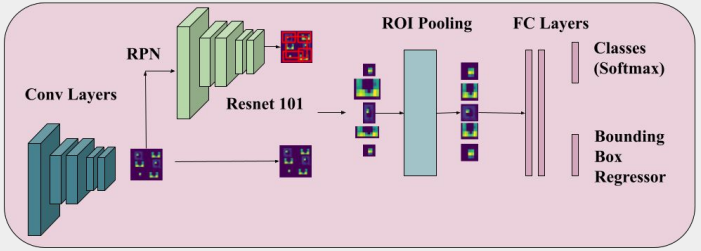
Remarks	Precision (in %)	Recall (in %)	F1-score (in %)
Trained on PBC 8 Cropped	94.78	94.77	94.79
Above model when used to classify Smear Slides 8 Cropped	4.47	21.29	7.45
Trained on Smear Slides 8 Cropped	82.64	82.25	81.74
Above model when used to classify PBC 8 Cropped	53.34	24.36	28.29
Model trained on mixed and classifying PBC 8 Cropped	95.21	95.45	95.37
Model trained on mixed and classifying Smear Slides 8 Cropped	83.98	83.73	84.34
Domain Adaptation Source: PBC Cropped Target: Smear Slides 8 Cropped, Testing on Smear Slides Cropped	78.89	78.31	77.11
Domain Adaptation Source: PBC Cropped Target: Smear Slides 8 Cropped, Testing on PBC 8 Cropped	95.64	95.77	95.89
Domain Adaptation Source: Smear Slides Cropped Target: PBC 8 Cropped, Testing on Smear Slides 8 Cropped	86.71	85.67	84.89
Domain Adaptation Source: Smear Slides 8 Cropped Target: PBC 8 Cropped, Testing on PBC 8 Cropped	93.11	93.21	93.37

Overall Architecture

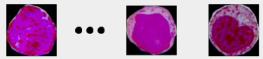
- The overall architecture of the pipeline is shown in Figure on the right.
- The initial slide is passed through the Mask RCNN, which takes care of extracting the cells, and giving initial layer of confidence to the classes.
- The cells are then passed to the 10-class classifier, which classifies the 10 classes. If the class is blast or band then the result is selected, else it is passed to the Domain Adaptation module to get the final classification.
- The Domain Adaptation module classifies the remaining 8 classes and gives an extra layer of confidence for the cells classes.

Input Smear Slide



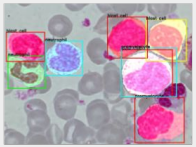


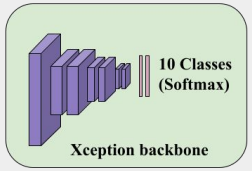
Mask RCNN



Extracted Cells

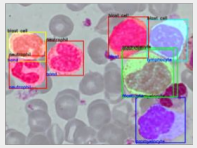
Mask RCNN Output with mask

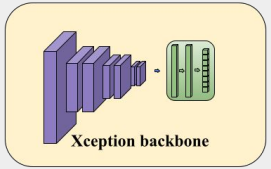




Classifier

Refined Output with mask

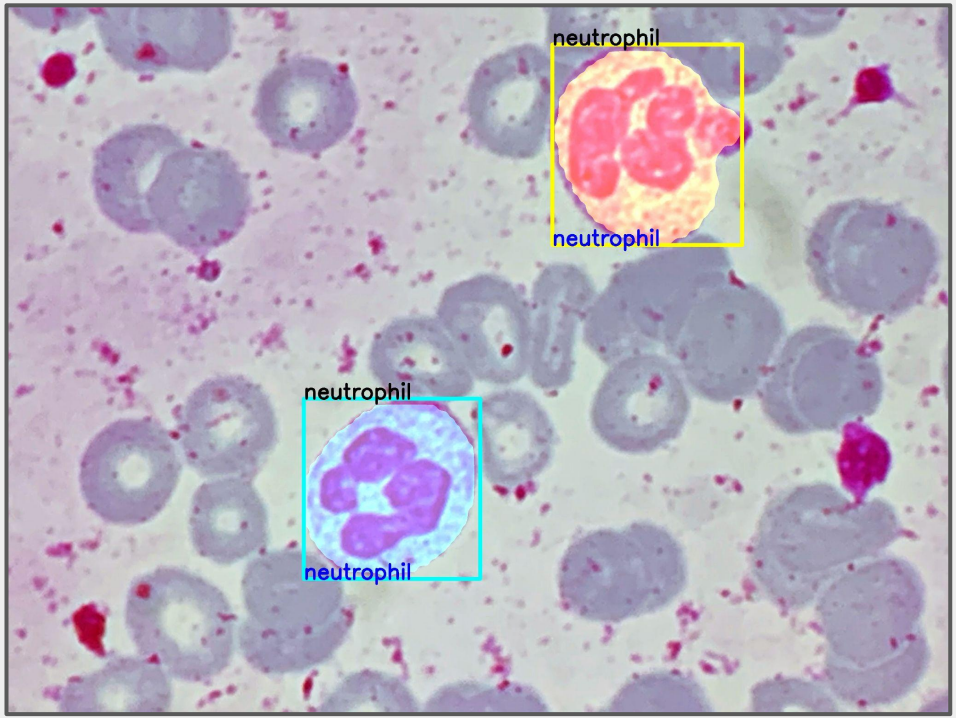


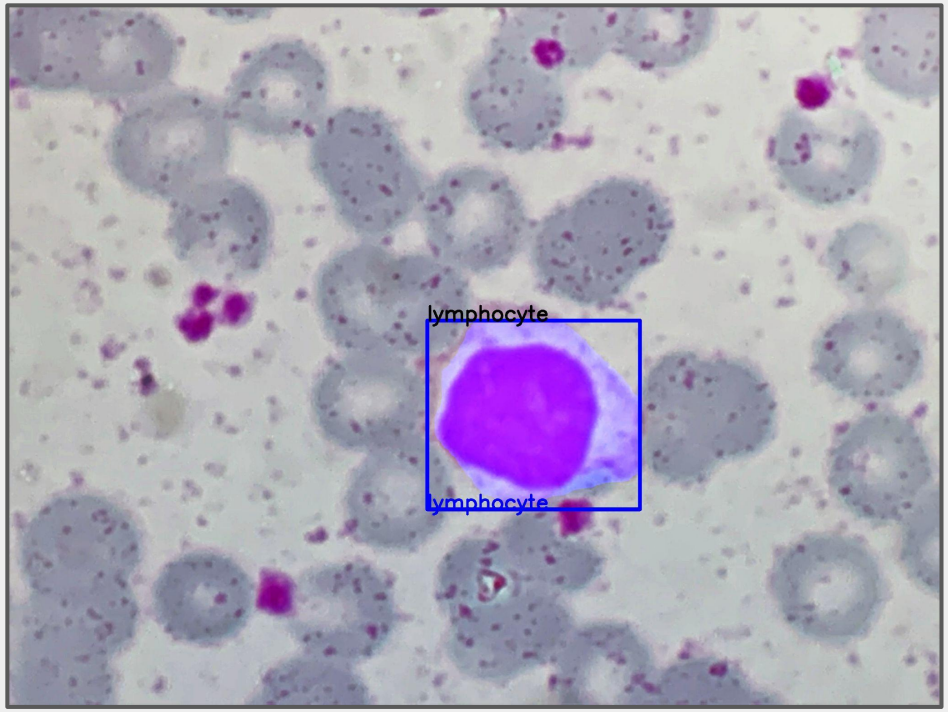


DA module with Classifier

Domain Adaptation Results - On Smear Slides Test Dataset (1)

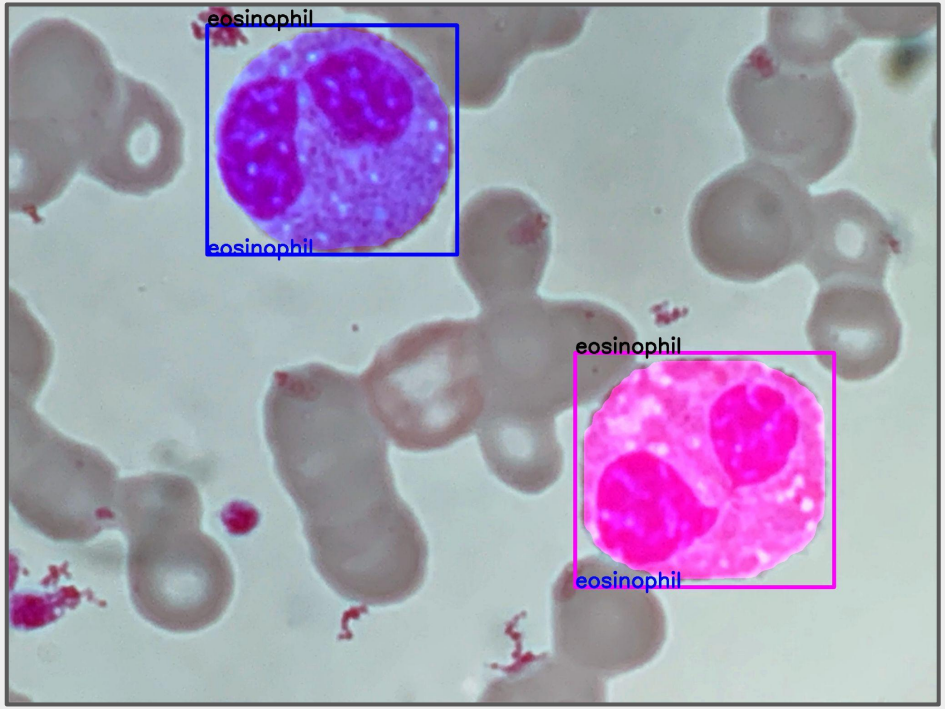
- The upper text is result from Mask RCNN and the lower one is Domain Adaptation's results.





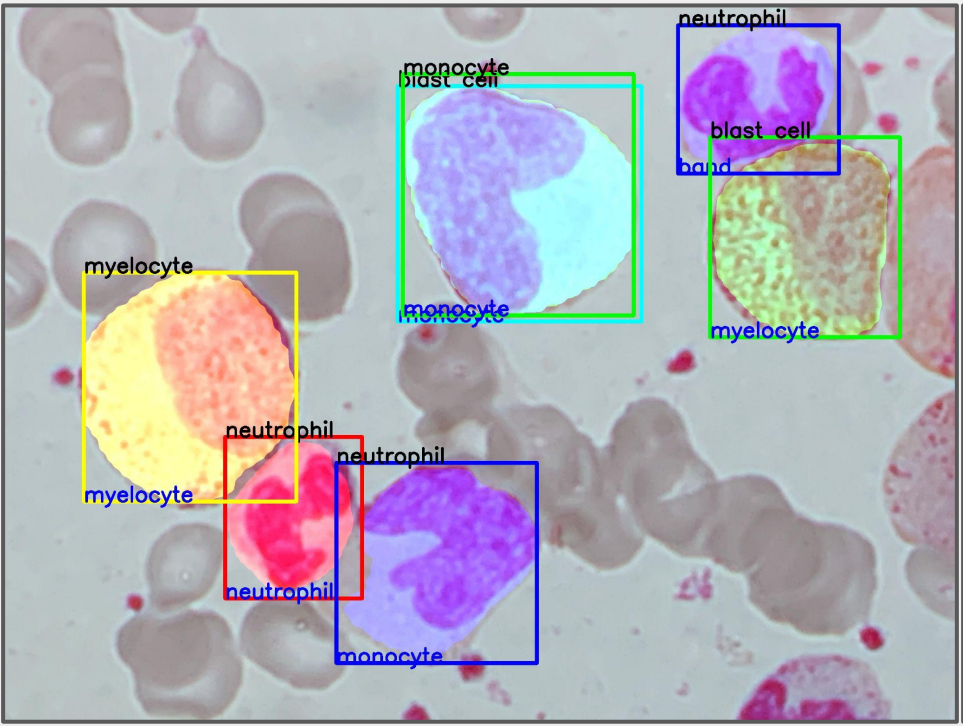
Domain Adaptation Results - On Smear Slides Test Dataset (2)

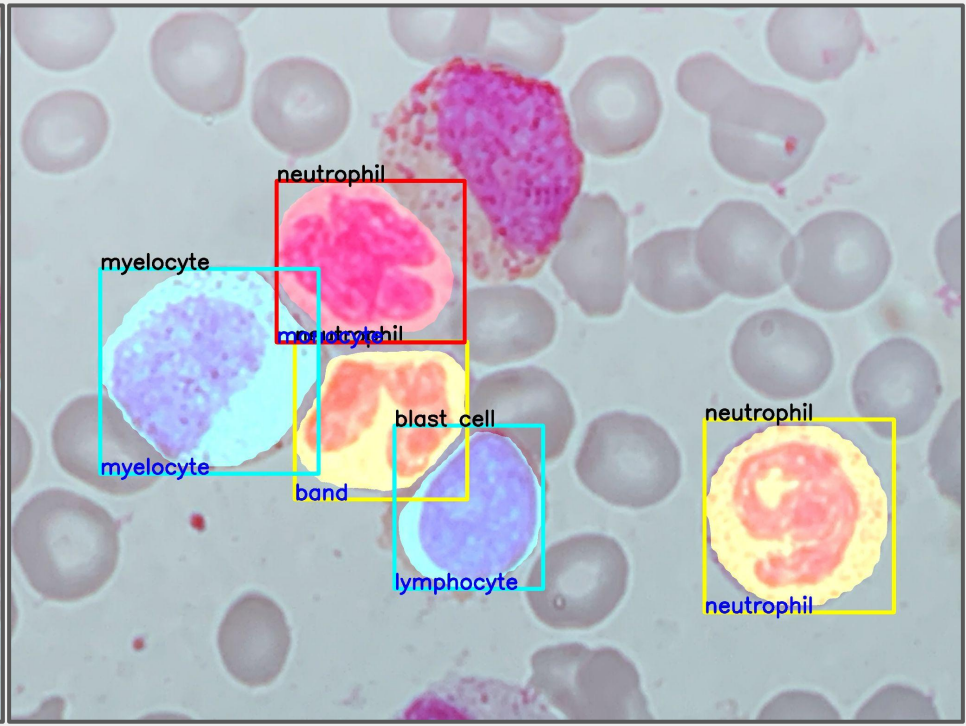
- The upper text is result from Mask RCNN and the lower one is Domain Adaptation results.



Domain Adaptation Results - On Smear Slides Mixed Dataset (2)

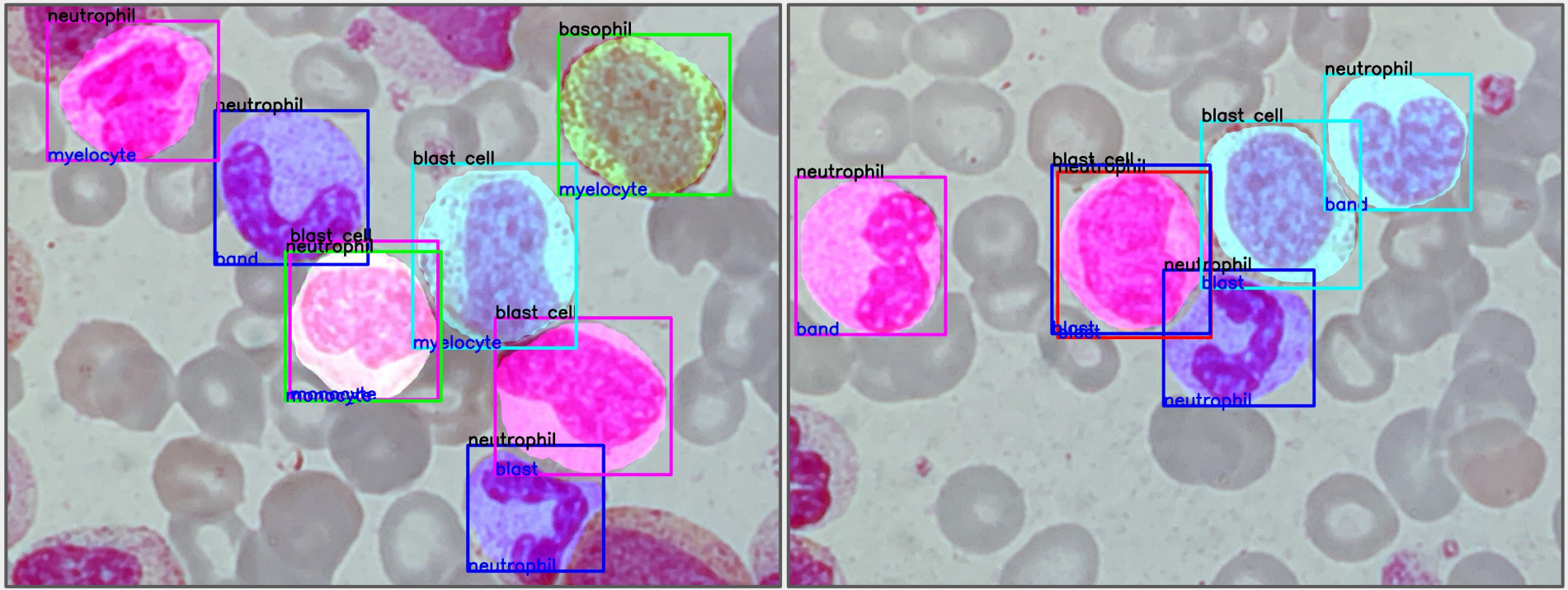
- The upper text is result from Mask RCNN and the lower one is Domain Adaptation results.





Domain Adaptation Results - On Smear Slides Mixed Dataset (1)

- The upper text is result from Mask RCNN and the lower one is Domain Adaptation results.



Conclusions

- A framework is created which can help medical practitioners for automating the process of Peripheral Blood Smear for diagnosing patients in future.

- Mask RCNN has been successfully applied to get the masks, and a script was created which could automate the process of masking using CVAT.

- A novel instance segmentation dataset is in the process of creation (583/5000) which could further spark new research, like generating synthetic data via GANs.

- Domain adaptation is used to minimize the misclassification as much as possible with the available data.

- Few experimental methods were conducted, like automating the segmentation using GRAD CAM which could be further improved in future.

References

1. Acevedo, Andrea; Merino, Anna; Alferez, Santiago; Molina, Ángel; Boldú, Laura; Rodellar, José (2020), “A dataset for microscopic peripheral blood cell images for development of automatic recognition systems”, Mendeley Data, V1, doi: 10.17632/snkd93bnjr.1.

2. Waleed Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask_RCNN, 2017.

3. Andrea Acevedo, Santiago Alférez, Anna Merino, Laura Puigví, and José Rodellar. Recognition of peripheral blood cell images using convolutional neural networks. Computer Methods and Programs in Biomedicine, 180:105020, 2019.

4. Serge Beucher. Watershed, hierarchical segmentation and waterfall algorithm. In Jean Serra and Pierre Soille, editors, Proceedings of the 2nd International Symposium on Mathematical Morphology and Its Applications to Image Processing, ISMM 1994, Fontainebleau, France, September 1994, volume 2 of Computational Imaging and Vision, pages 69–76. Kluwer, 1994.

5. François Chollet. Xception: Deep learning with depthwise separable convolutions. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1800–1807, 2017.

References

6. Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15, page 1180–1189. JMLR.org, 2015.

7. Nazli Farajidavar. Transductive transfer learning for computer vision., 2015.

8. Hao Guan and Mingxia Liu. Domain adaptation for medical image analysis: A survey, 2021.

9. Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In 2017 IEEE International Conference on Computer Vision (ICCV), pages 2980–2988, 2017.

10. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, 2016.

11. Han-Kai Hsu, Wei-Chih Hung, Hung-Yu Tseng, Chun-Han Yao, Yi-Hsuan Tsai, Maneesh Singh, and Ming-Hsuan Yang. Progressive domain adaptation for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2019.

References

12. Genta Ishikawa, Rong Xu, Jun Ohya, and Hiroyasu Iwata. Detecting a fetus in ultrasound images using grad CAM and locating the fetus in the uterus. In Maria De Marsico, Gabriella Sanniti di Baja, and Ana L. N. Fred, editors, Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods, ICPRAM 2019, Prague, Czech Republic, February 19-21, 2019, pages 181–189. SciTePress, 2019.

13. I. T. Jolliffe. Principal Component Analysis and Factor Analysis, pages 115–128. Springer New York, New York, NY, 1986.

14. Michael Linden, Jerrold M. Ward, and Sindhu Cherian. Hematopoietic and Lymphoid Tissues, pages 309–338. Elsevier Inc., 2012. Copyright: Copyright 2013 Elsevier B.V., All rights reserved.

15. Shuying Liu and Weihong Deng. Very deep convolutional neural network based image classification using small training sample size. In 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), pages 730–734, 2015.

16. Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4500–4509, 2018.

References

17. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6):1137–1149, 2017.

18. Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In 2017 IEEE International Conference on Computer Vision (ICCV), pages 618–626, 2017.

19. Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inceptionv4, inception-resnet and the impact of residual connections on learning, 2016.

20. Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2818–2826, 2016.

21. Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2962–2971, 2017.

References

22. Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. Journal of Machine Learning Research, 9(86):2579–2605, 2008.

23. Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, Computer Vision – ECCV 2014, pages 818–833, Cham, 2014. Springer International Publishing.

24. Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8697–8710, 2018.

25. H. Robbins. A stochastic approximation method. Annals of Mathematical Statistics, 22:400–407, 2007.

26. Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, Computer Vision – ECCV 2014, pages 740–755, Cham, 2014. Springer International Publishing.

Acknowledgements

I am thankful to Tamal Mj for his guidance and close scrutiny.

I would also like to acknowledge the help and support received by Aniket Bhattacharyea.

Finally, I would like to express my deepest gratitude to Swathy Prabhu Mj for generously arranging the compute resources (i.e., Asus RTX **2080** Ti (**12** GB) and Quadro GV100 (**32** GB) GPUs with **64** GB RAM, to hasten the research), and also to Dripta Mj for the stimulating discussions that he had shared with me during the Advanced Machine Learning course.

I am grateful to my father, Dr. Jadab Kumar Pal, Deputy Chief Executive, Indian Statistical Institute, Kolkata for constantly motivating and supporting me to develop this documentation along with the proofreading. I will also mention about my brother Jisnoo Dev Pal and my mother Sumita Pal, for supporting me.

Thank You. Any questions?

jimutbahanpal@yahoo.com