THE ANNUAL TECHNO-MANAGEMENT FEST, IIT(BHU) VARANASI

TECHNEX'22

11-13 MAR 2022

ML-Ware'22

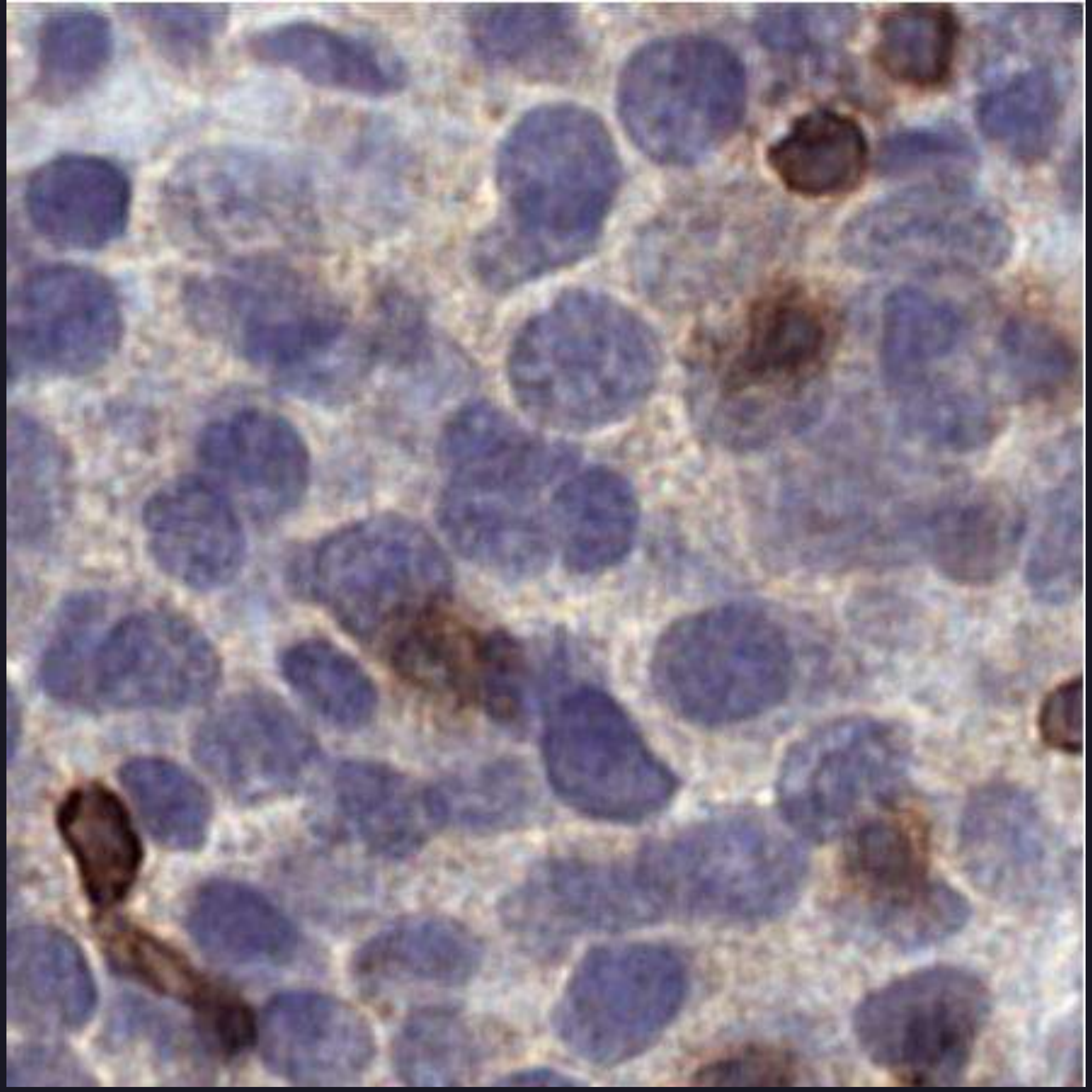
Counting stained cells from microscopic imagery

By

Srikar Verma

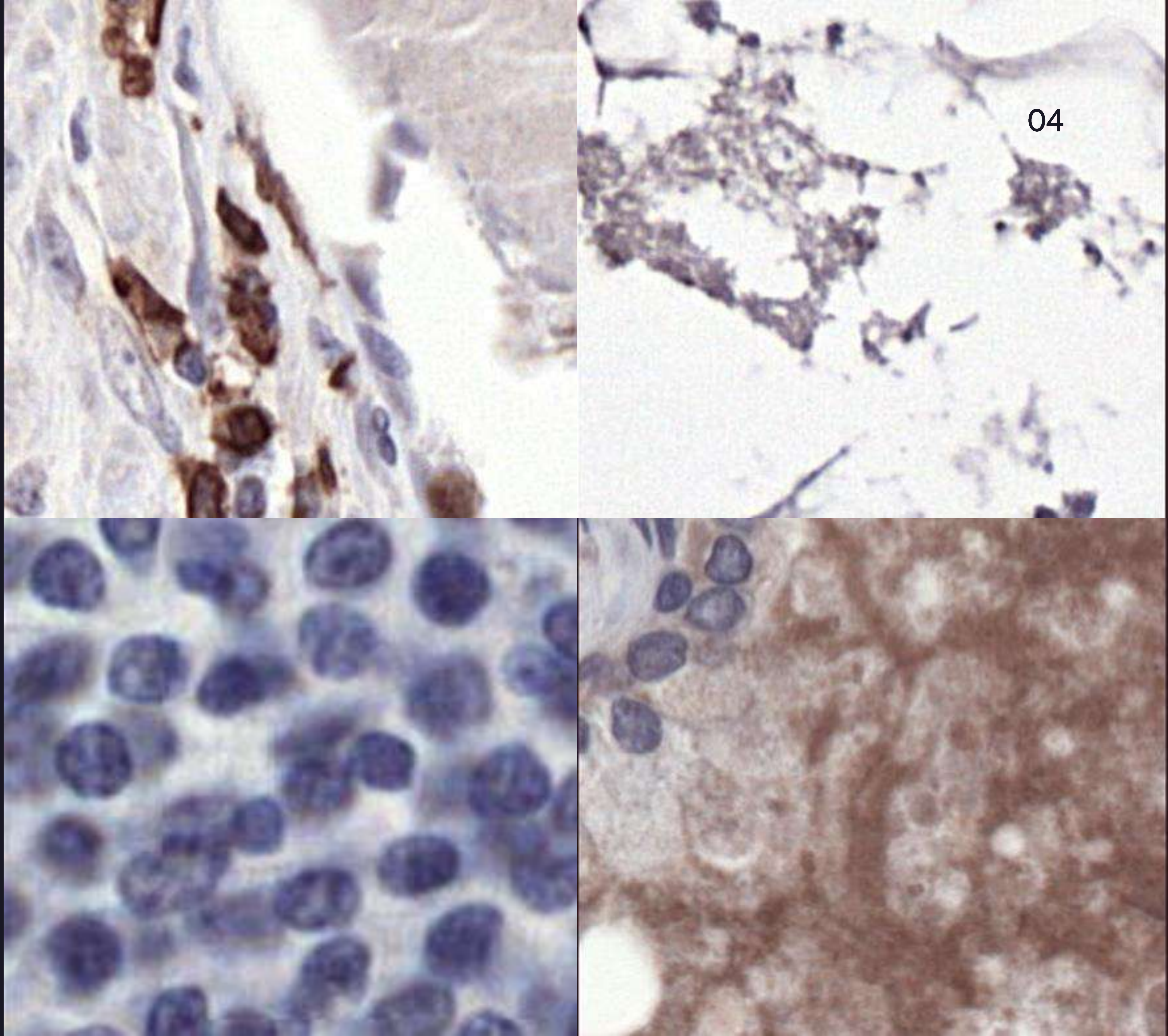
Kaggle ID - jarvis

Problem Statement



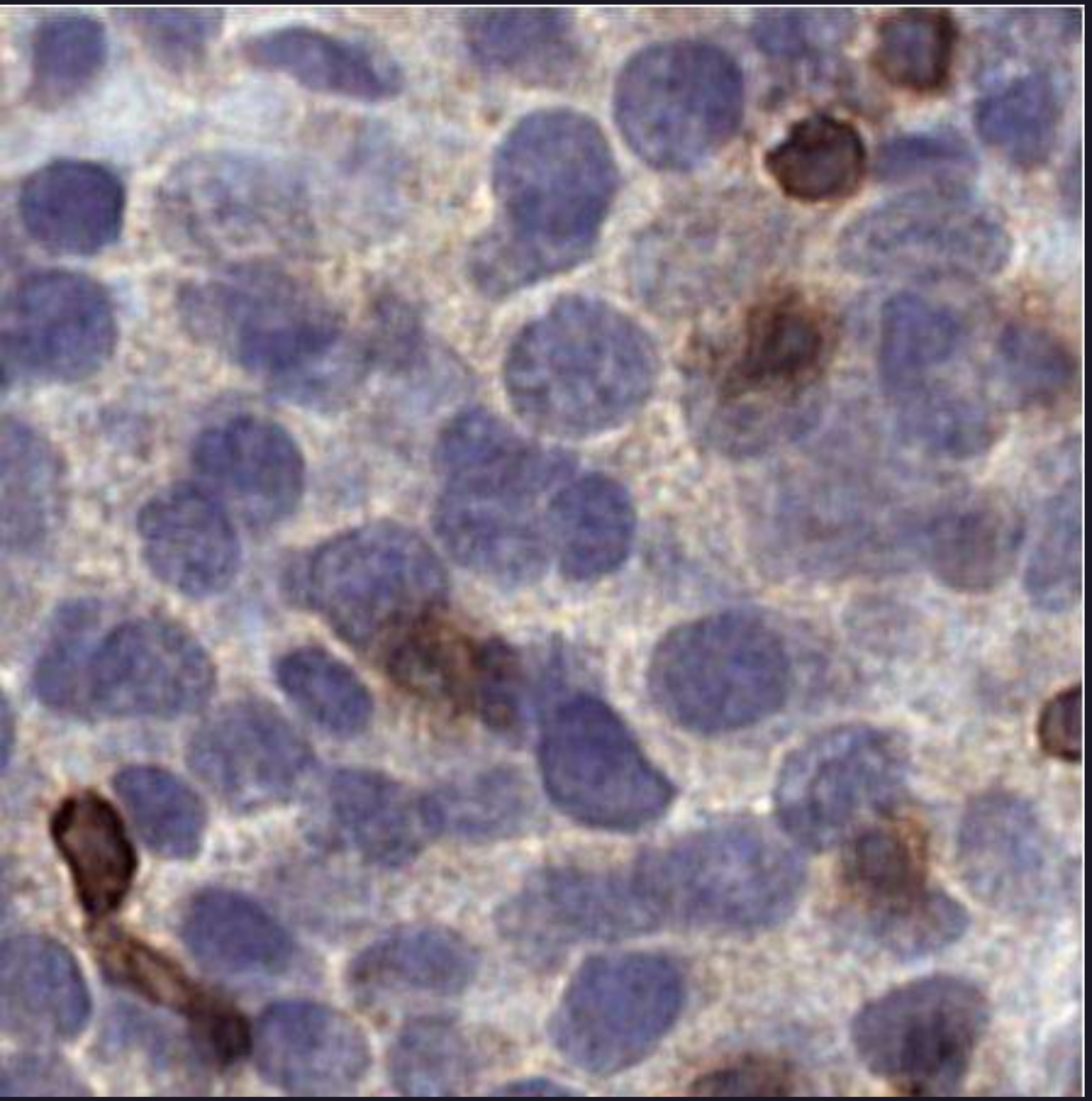
Count the number of stained cells with a blue nucleus and a brown membrane, in the given miroscopic images.

Exploring the data



04

Data and Preprocessing



Training data :

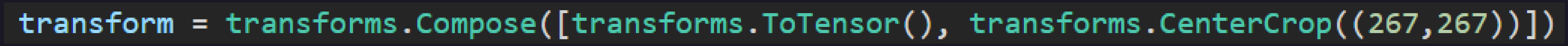
13,400 microscopic images - size 299x299x3

Labels - number of stained cells with blue nucleus and brown membrane.

Image preprocessing:

Pixel values normalized in the range (0,1)

A border of 16 pixels cropped.



```
transform = transforms.Compose([transforms.ToTensor(), transforms.CenterCrop((267,267))])
```

Potential Solutions

Machine Learning Pipeline

A pipeline made up of a feature extraction method, and a regression method.
However, combining these might be a complex task, and we can also use a single deep learning pipeline as an alternative.

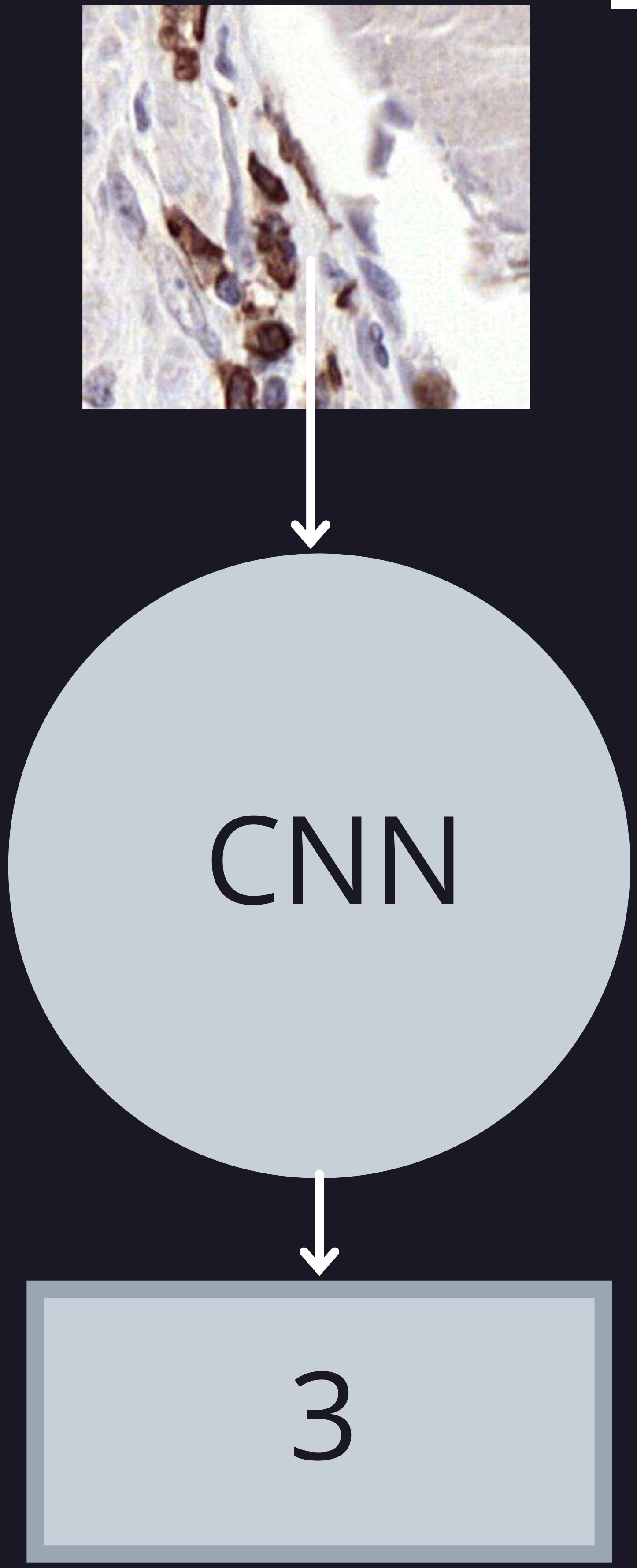
Detection or Segmentation

Segmentation/Detection models cannot be trained due to lack of annotated data.
Also, these models would perform poorly in counting cells in a high cell density region.

Using CNNs for Regression

We could use Convolutional Network architectures, and modify them for regression tasks, so that it could learn a mapping from image input to a numeric output, denoting the number of cells.

Finalizing the Approach


CNN

3

To develop a cell counter, the problem is considered to be a regression task, where we are interested in learning the relationship between image representations and the number of cells.

We may use a Convolutional Neural Network (CNN) that directly predicts the number of cells, i.e., a regression model.

Building the Regression Model

Selecting the Model

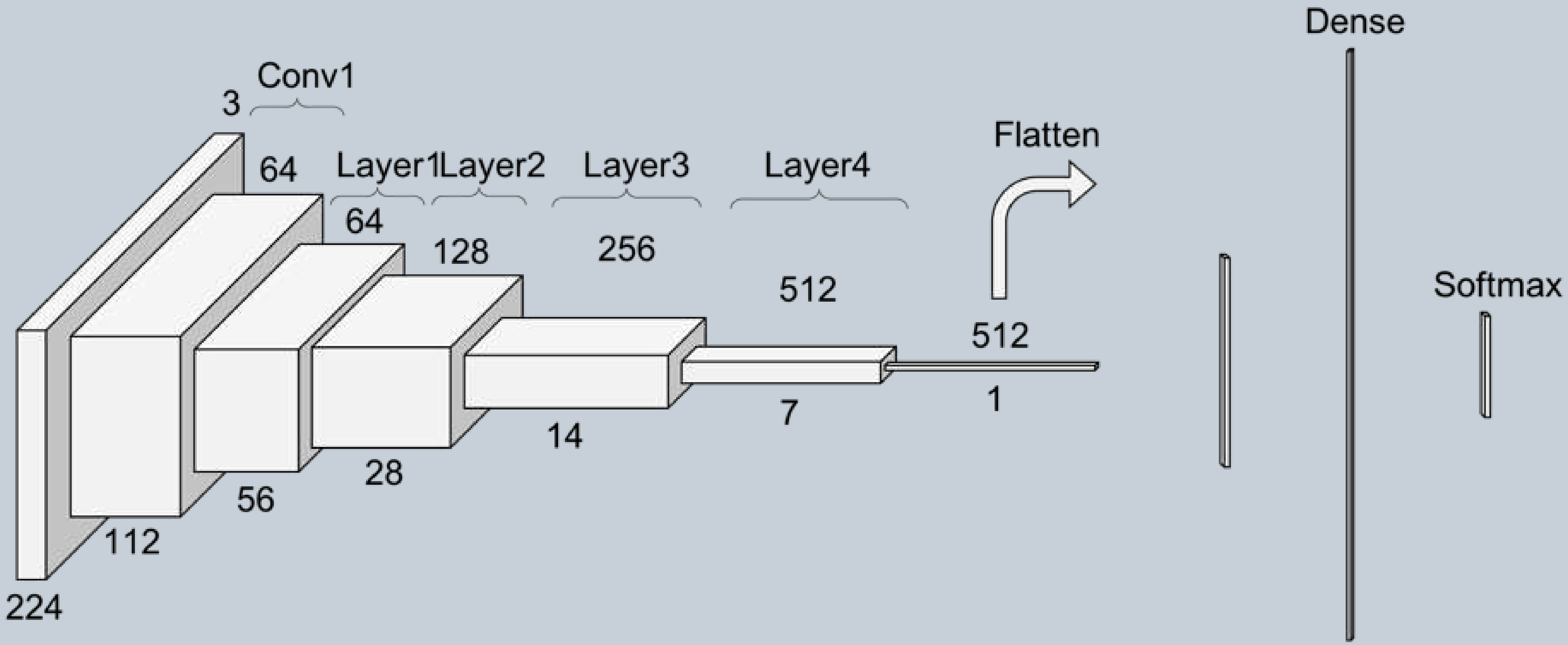
- As our task required counting specific cells from microscopic images, a deep CNN architecture can be used, which could accurately map images to cell count.

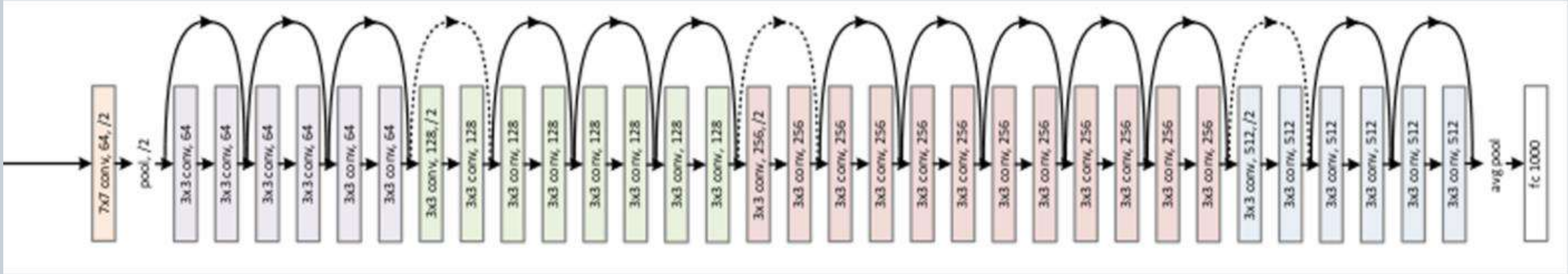
- ResNets are powerful CNN-based classification models, which can accurately classify upto 1000 classes of objects.

- The ResNet model architecture was selected, due to relatively simple architecture, and the use of skip connections in its layers which significantly improves accuracy.

- After trying different variants of ResNet, including ResNet-18, ResNet-34, and ResNet-50, ResNet-34 performed best amongst them, and was thus finalized.

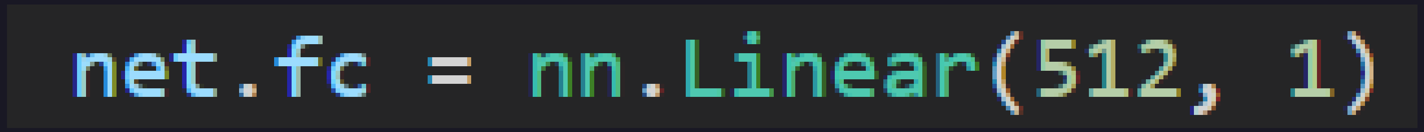
ResNet-34 Architecture





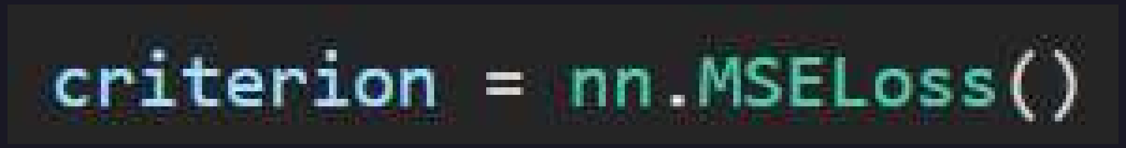
ResNet-34 for Cell Counting

- ResNet-34 model is originally used for multiclass classification tasks.
- But, as we use the model for regression, the output layer of the model, was changed from 1000 cells to 1 cell.

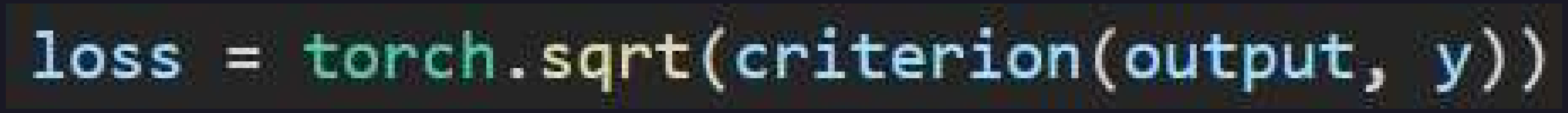


```
net.fc = nn.Linear(512, 1)
```

- For the regression task, the RMSE Loss function was used.

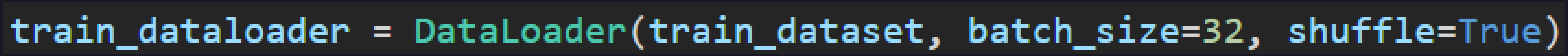


```
criterion = nn.MSELoss()
```



```
loss = torch.sqrt(criterion(output, y))
```

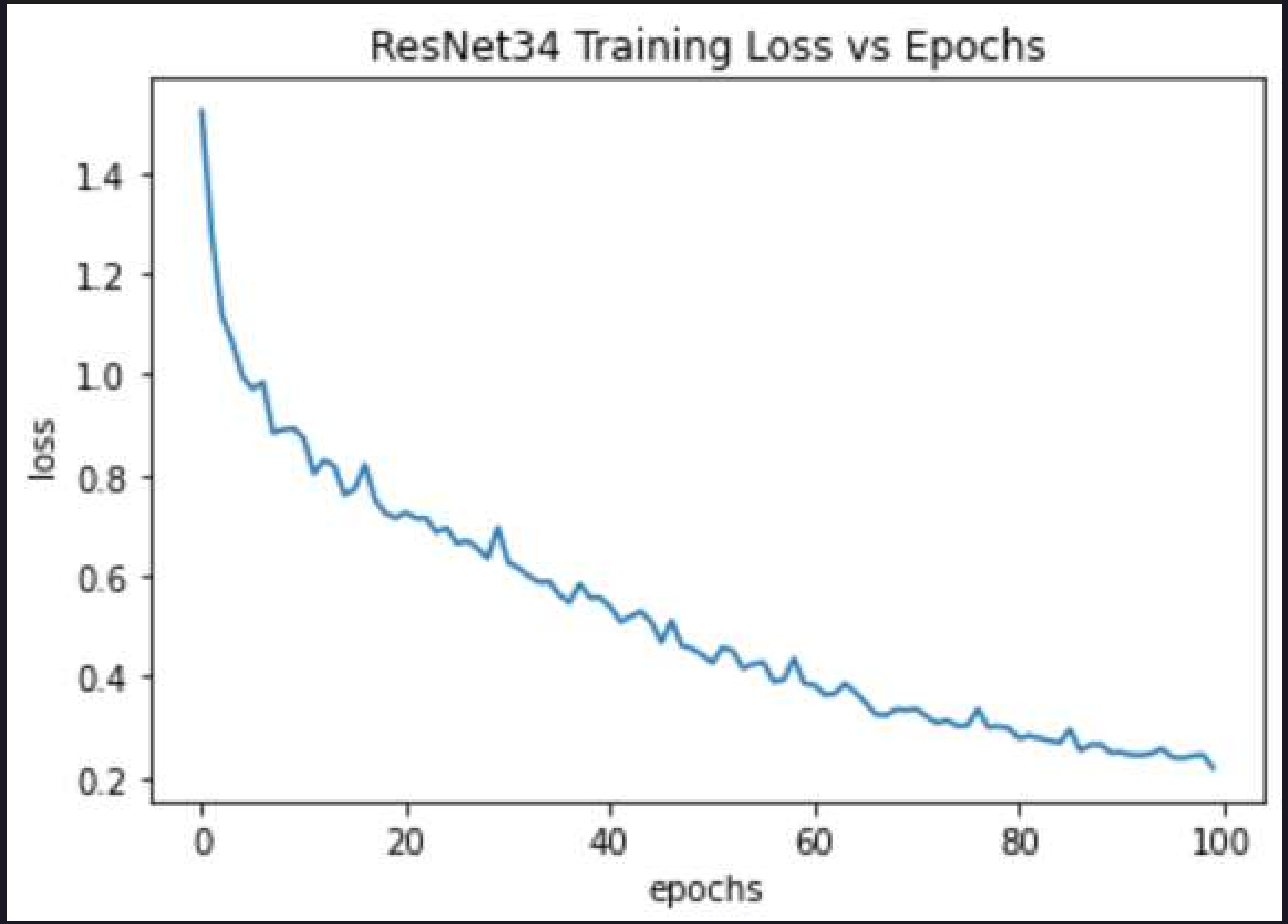
- The model was trained on batches of size 32, for 100 epochs.



```
train_dataloader = DataLoader(train_dataset, batch_size=32, shuffle=True)
```

- Adam optimizer was used, with a learning rate 5e-5.

Results



- Predictions were made on the test images using the model.
- RMSE Score on public leaderboard - 1.15
- RMSE Loss on private leaderboard - 0.98

Conclusion

- Here, we conclude that, even a complex task like counting the number of very specific cells (objects), can be accomplished using Convolutional Neural Networks like ResNet modified for regression.

- We can also conclude that, multiclass image classification models, specially ResNets, can also be suited for regression tasks, and their performance can be as commendable as that for classification tasks.

- We finally conclude that the ResNet-34 model modified for regression performed a well on the microscopic image data, achieving a good score and low RMSE.

THANK YOU!