

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Blood Cell Classification Using Deep CNN Architectures

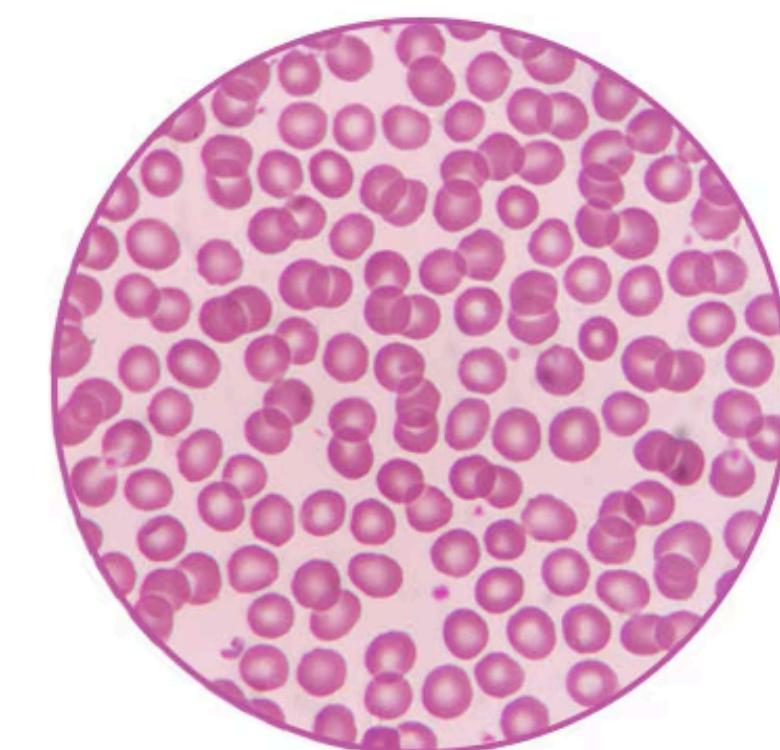
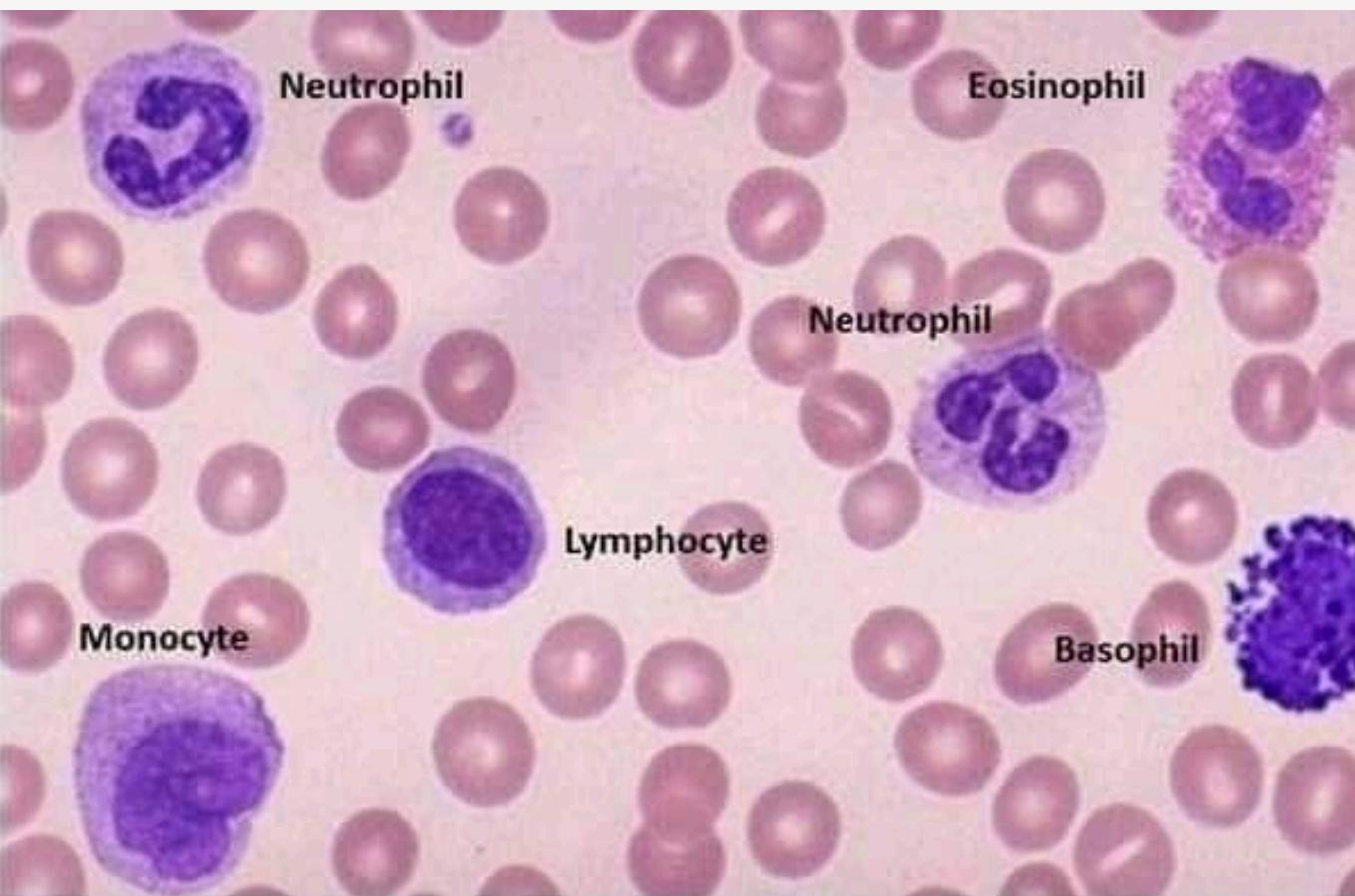
Machine Learning For Human Data , A.Y. 2024/2025

Işıkay Karakuş
2071938

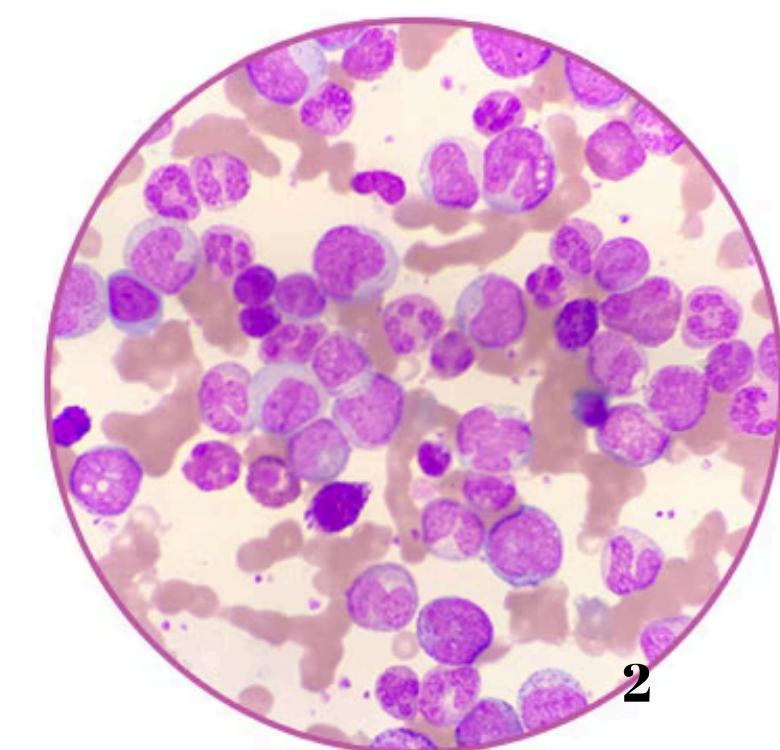
Introduction to Blood Cell Classification



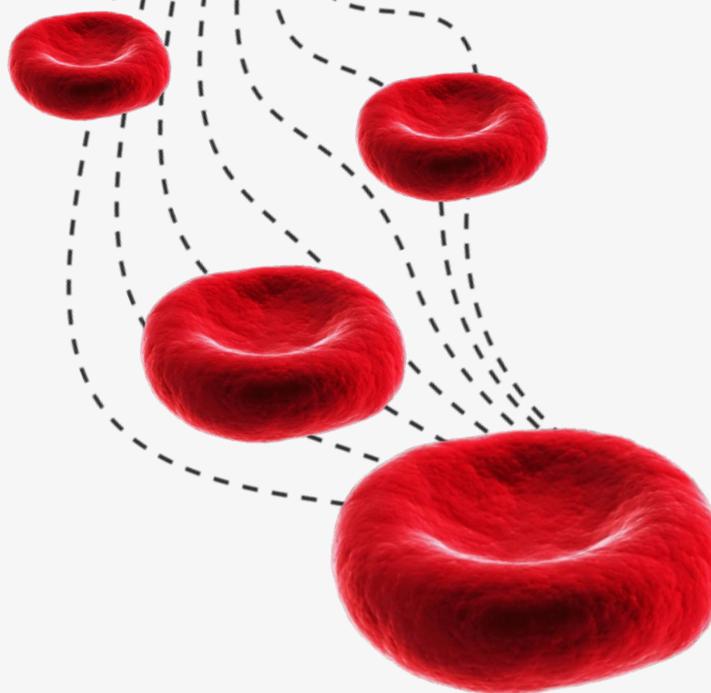
Peripheral blood cell classification is a critical step in medical diagnostics, helping identify infections, cancers, and immune disorders through microscope analysis.



Normal Blood



Leukaemia Blood



Challenges

01

Subtle visual differences between cell types

- Monocytes vs. Immature Granulocytes

02

Class imbalance in the dataset

- Some classes are rare, making learning biased

03

Image variability due to lab conditions

- Staining, lighting, or equipment may differ

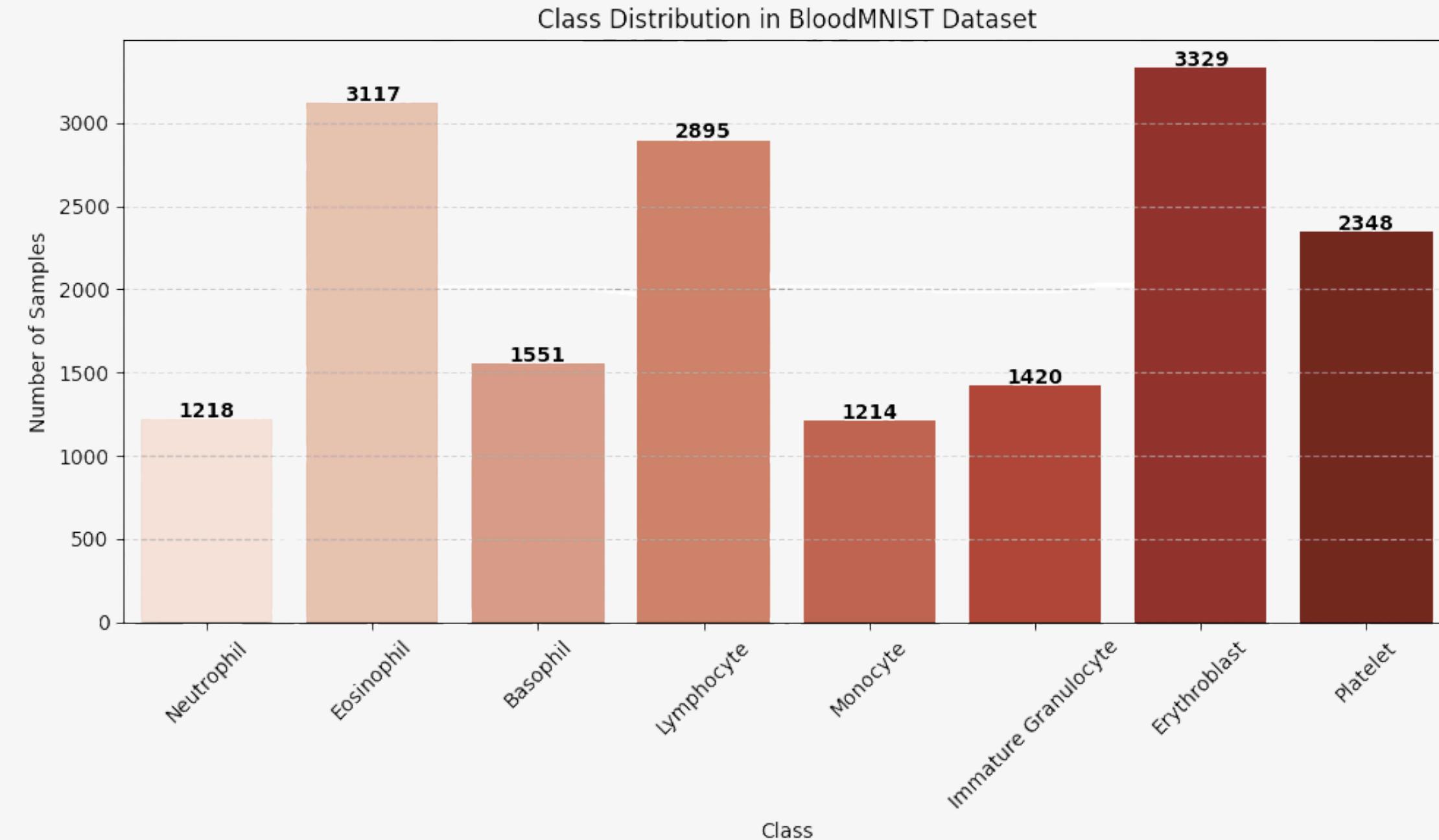
04

Clinical-grade accuracy required

- Even small errors can lead to misdiagnosis

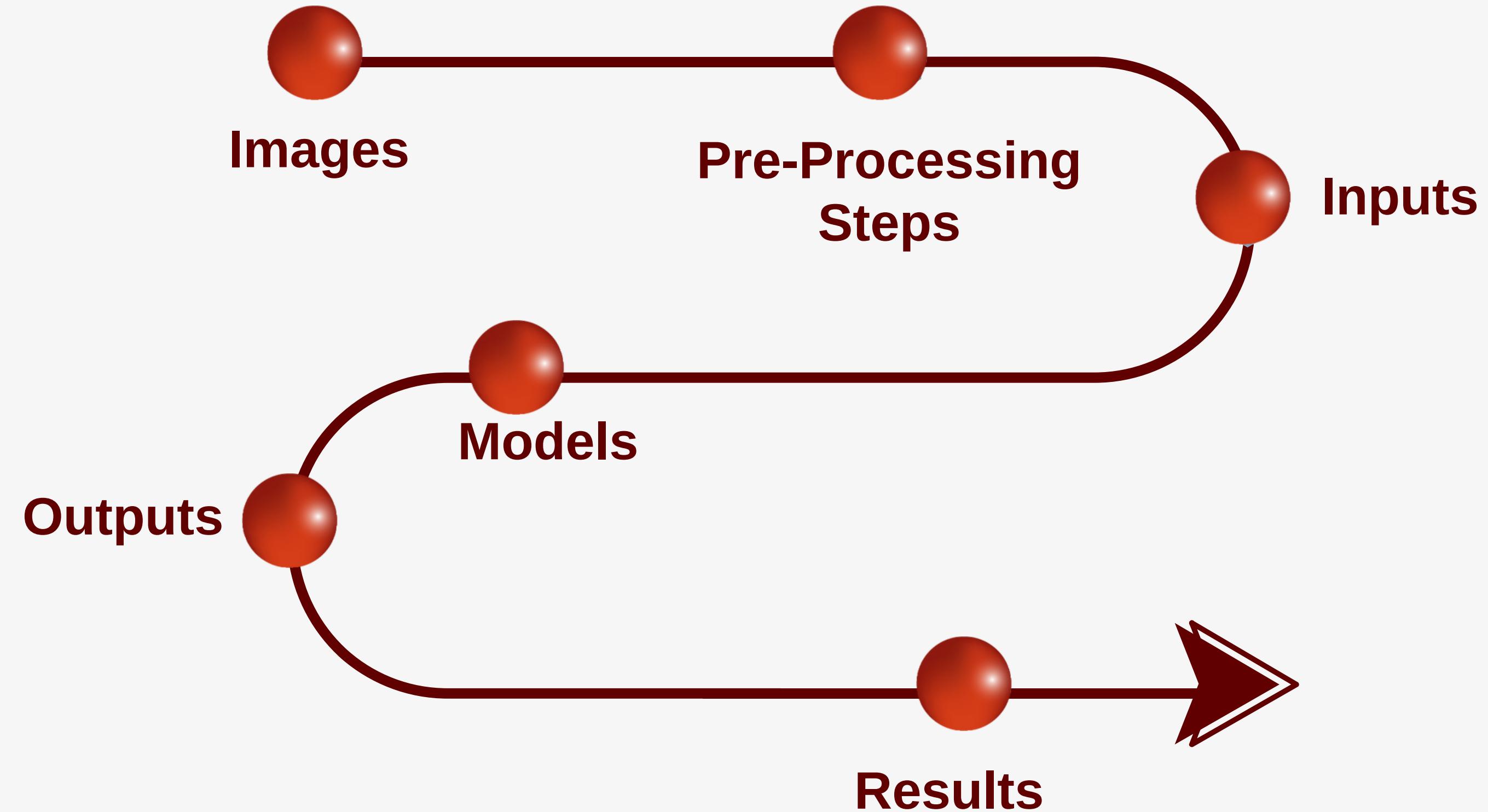
Machine and Deep Learning models can help automate and improve accuracy with consistent and scalable predictions

Dataset Introduction - BloodMNIST



- Total samples: 17092
- Number of classes: 8 (Multi-Class)
- Part of the MedMNIST v2 collection
- <https://zenodo.org/records/10519652>
- Three versions: image sizes 3x64x64, 3x128x128, and 3x224x224

Processing Pipeline



End-to-end workflow for training and evaluating CNN architectures on BloodMNIST

Pre-Processing Steps

Dataset Resolutions

- 64×64 version: for the baseline CNN model
- 128×128 version: for advanced CNN models
- Pre-resized datasets provided by BloodMNIST

Dataset Splits

- Used predefined train/validation/test splits:
 - **70% training** (11,959 images)
 - **10% validation** (1,712 images)
 - **20% test** (3,421 images)

Class Weighting for Imbalance

- Used *Scikit-learn's* `compute_class_weight('balanced')`
- Passed class weights to the loss function to handle imbalance during training

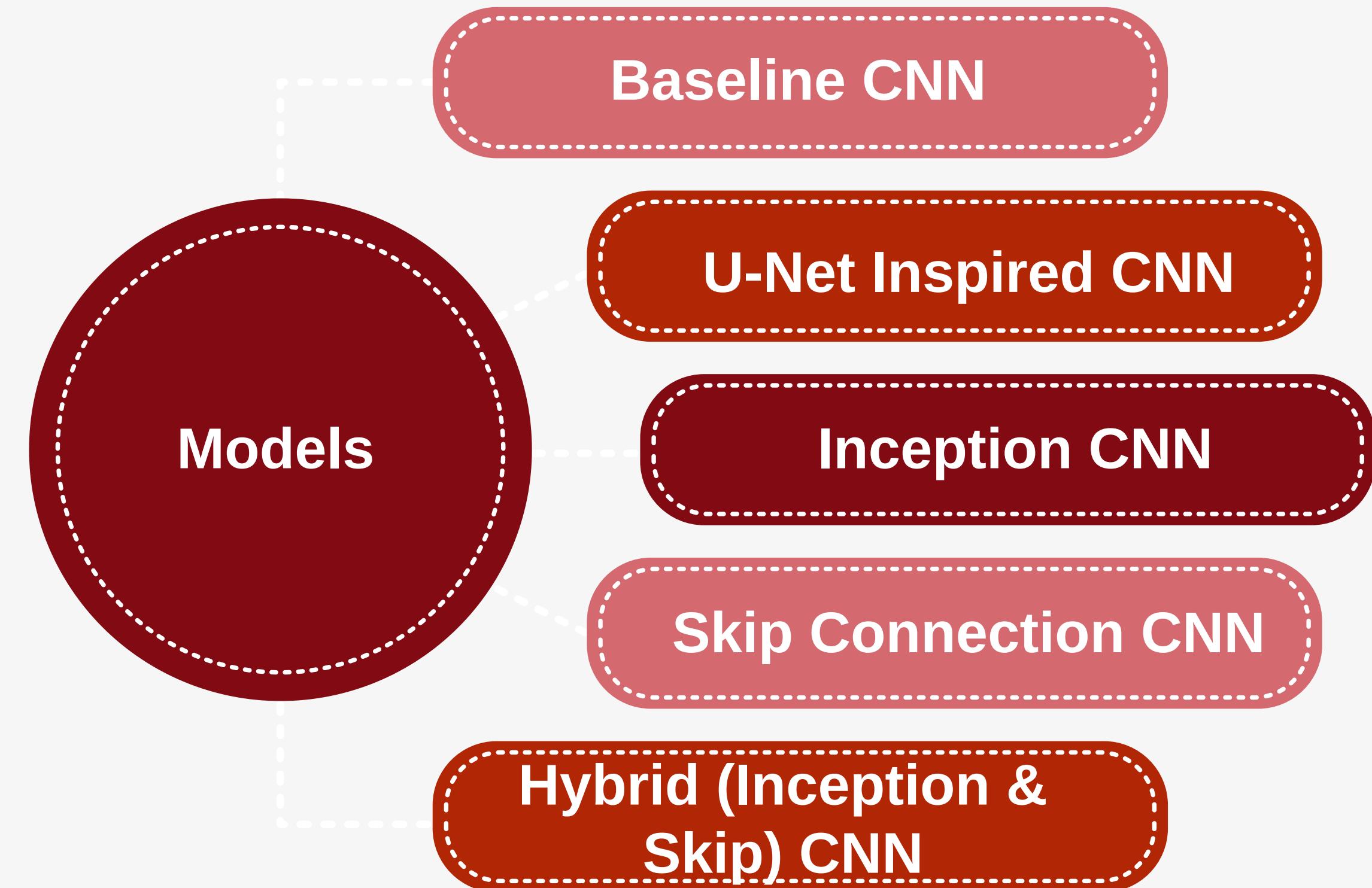
Normalization

- Normalized pixel values to [0, 1] by dividing all RGB channels by 255

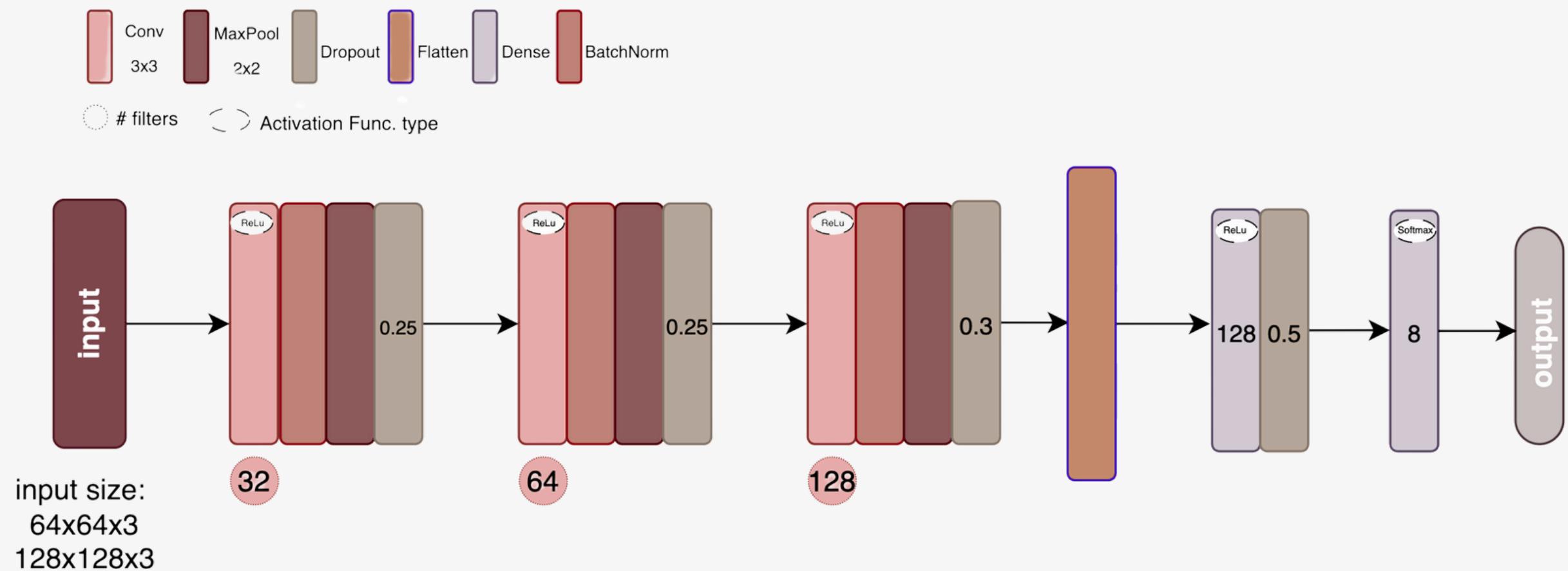
Data Augmentation

- Applied only to *training data*
- Random rotations (0°, 90°, 180°, 270°)
- Horizontal flipping
- Zoom shift via random cropping (retaining 90–100% of original area), then resizing back to 128×128

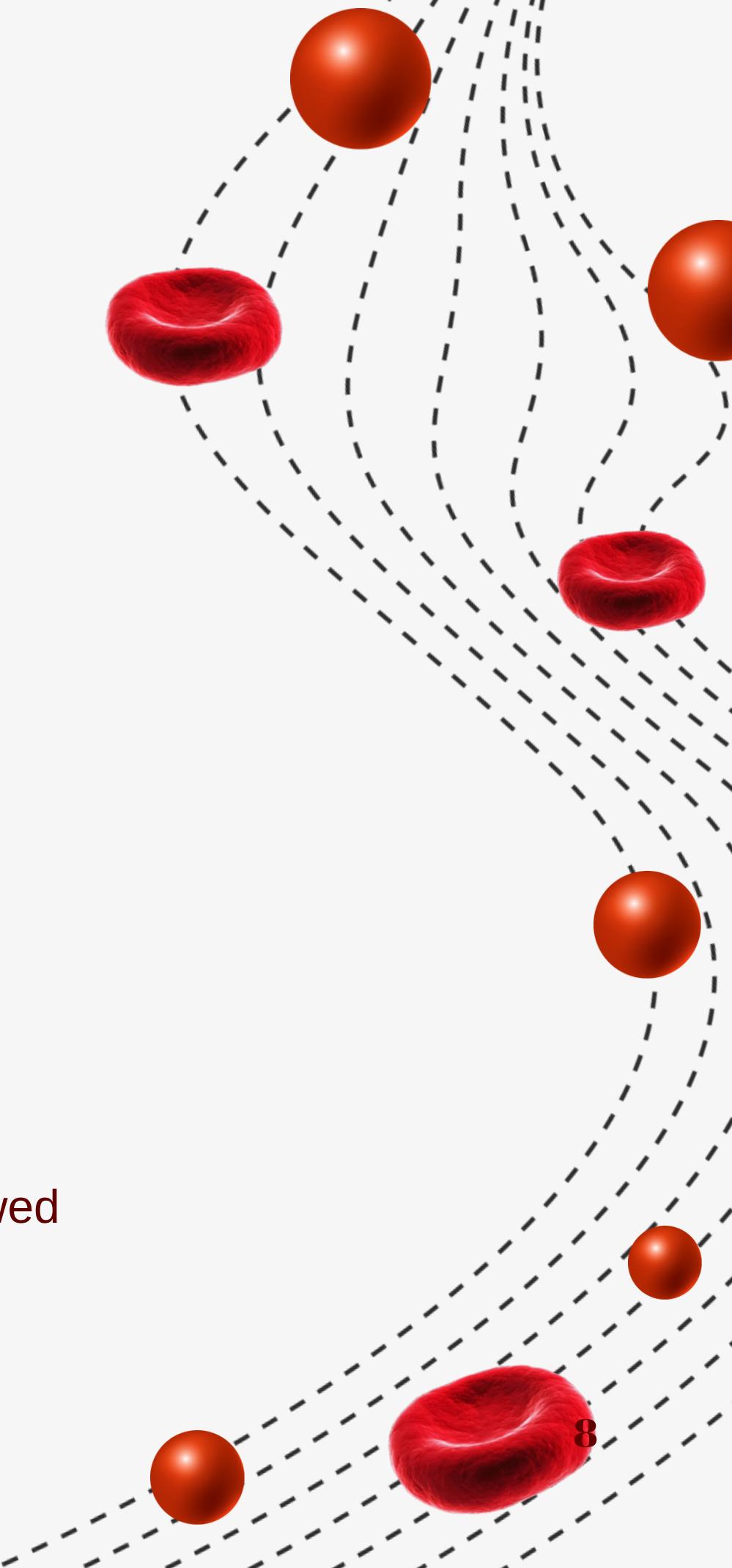
Models Overview



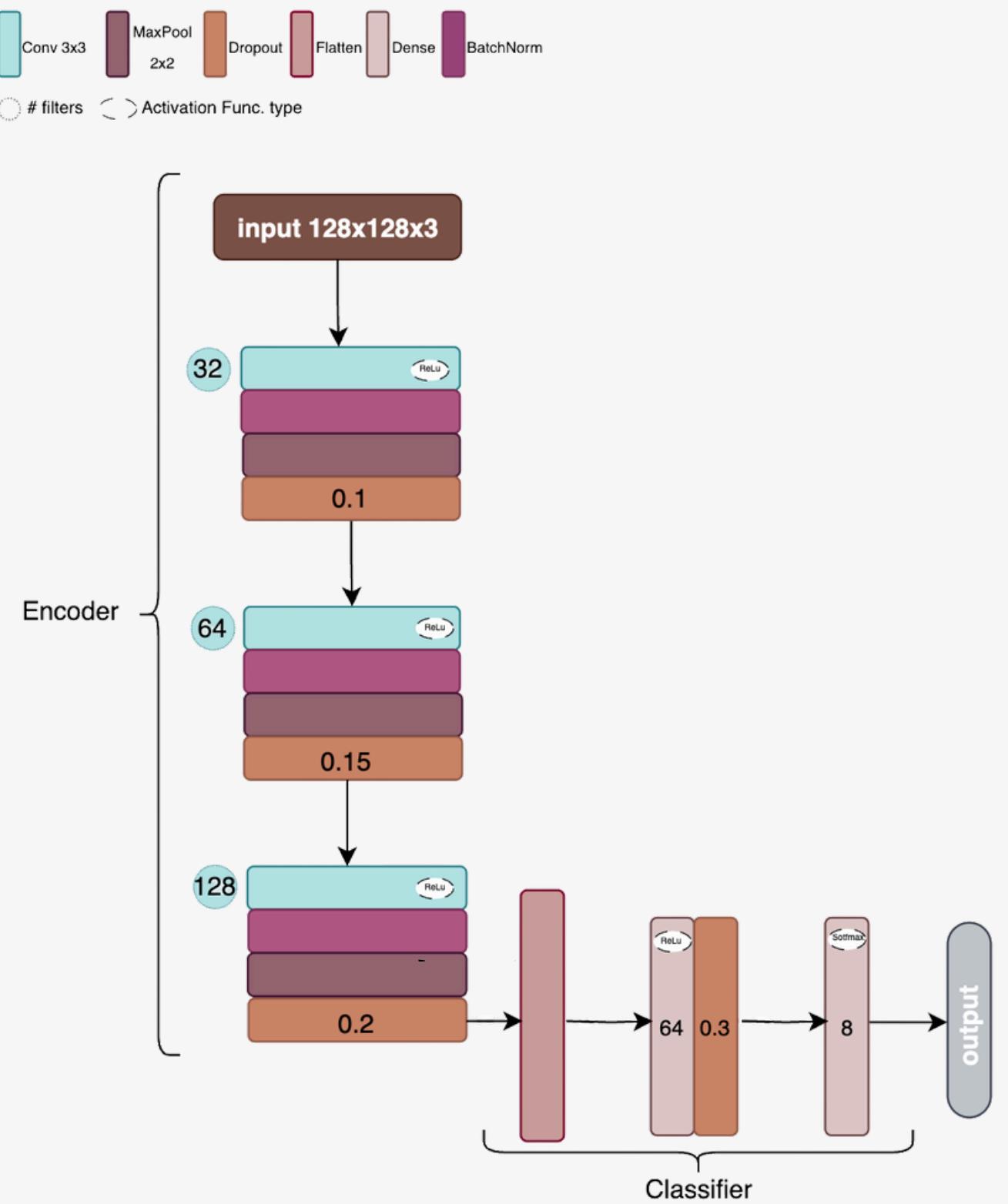
Baseline CNN



- **Architecture:** 3 convolutional blocks with increasing filters (32–64–128), each followed by BatchNorm, MaxPooling, and Dropout
- **Classifier:** Flatten \rightarrow Dense(128) \rightarrow Dropout(0.5) \rightarrow Softmax(8)
- **Input sizes tested:** $64 \times 64 \times 3$ and $128 \times 128 \times 3$
- **Regularization:** Dropout (0.25–0.5) and BatchNorm
- **Goal:** Serves as a reference model to benchmark deeper architectures

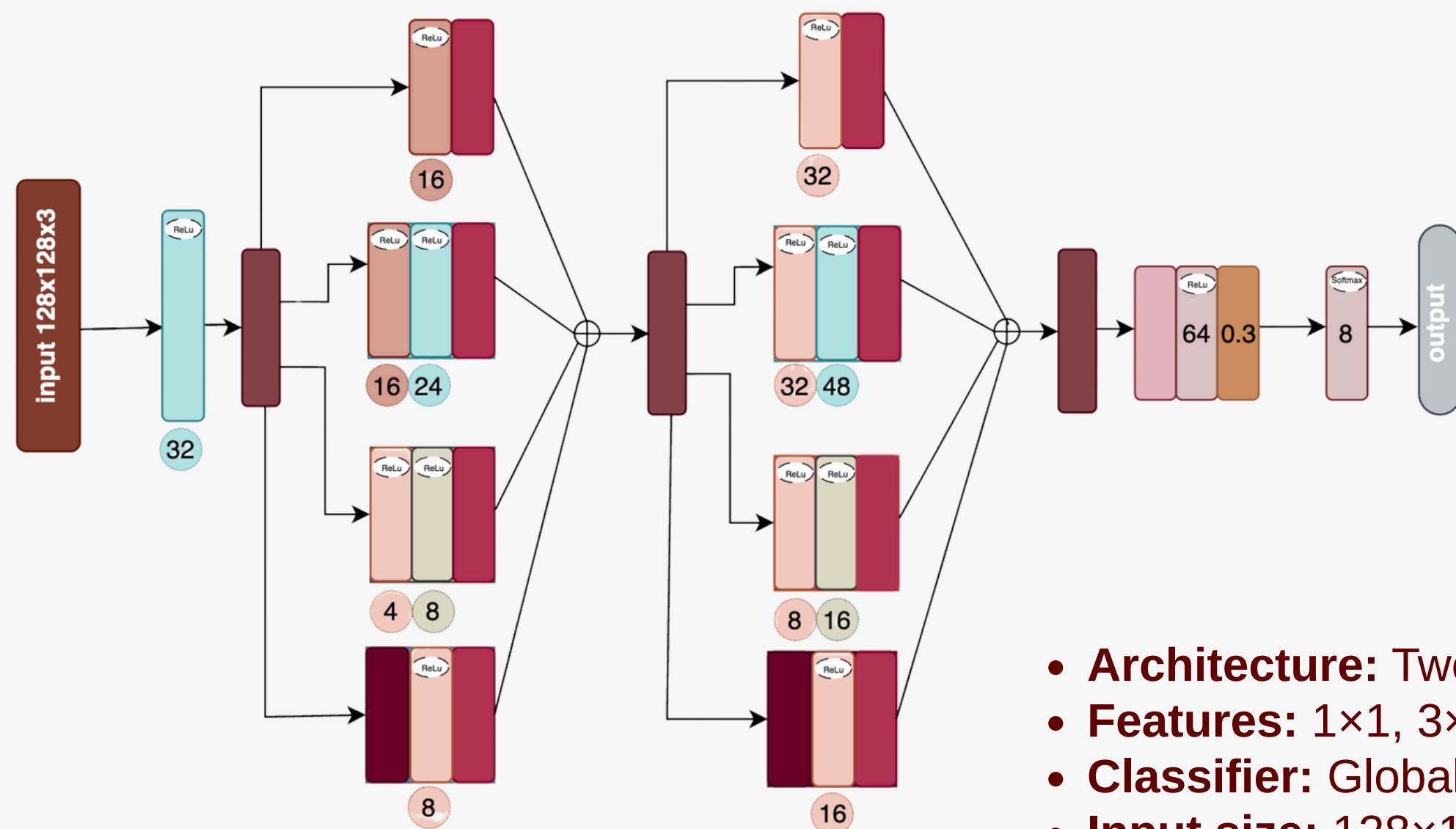
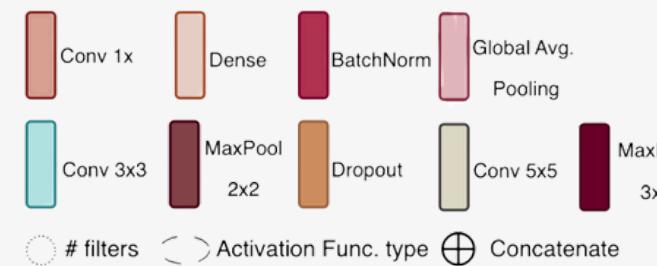


U-Net Encoder Inspired CNN



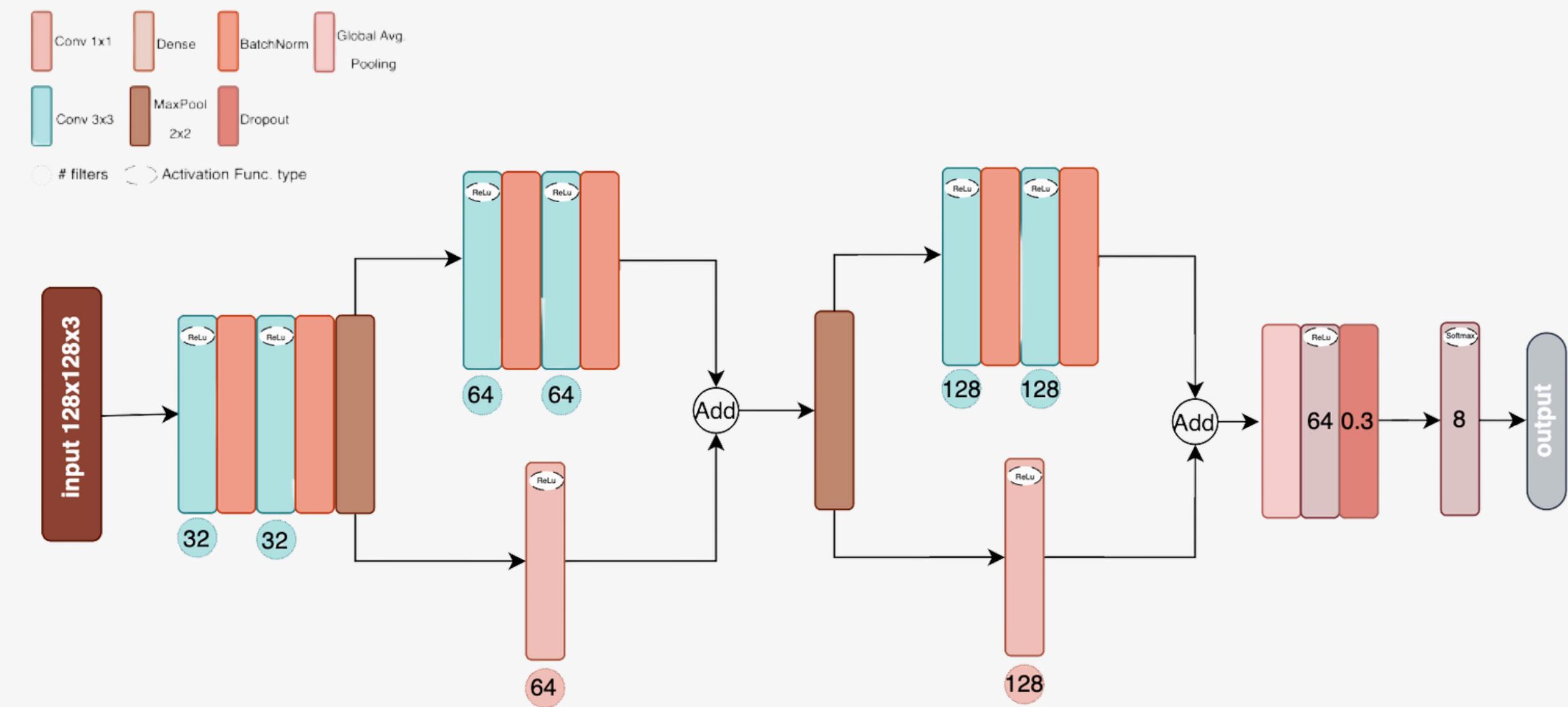
- **Architecture:** Encoder-only U-Net design
- **Regularization:** Dropout (0.1–0.2) and BatchNorm
- **Input size:** 128×128×3
- **Classifier:** Flatten → Dense(64) → Dropout(0.3) → Output(8)
- **Goal:** Compact deep feature extractor with fewer parameters

Inception CNN

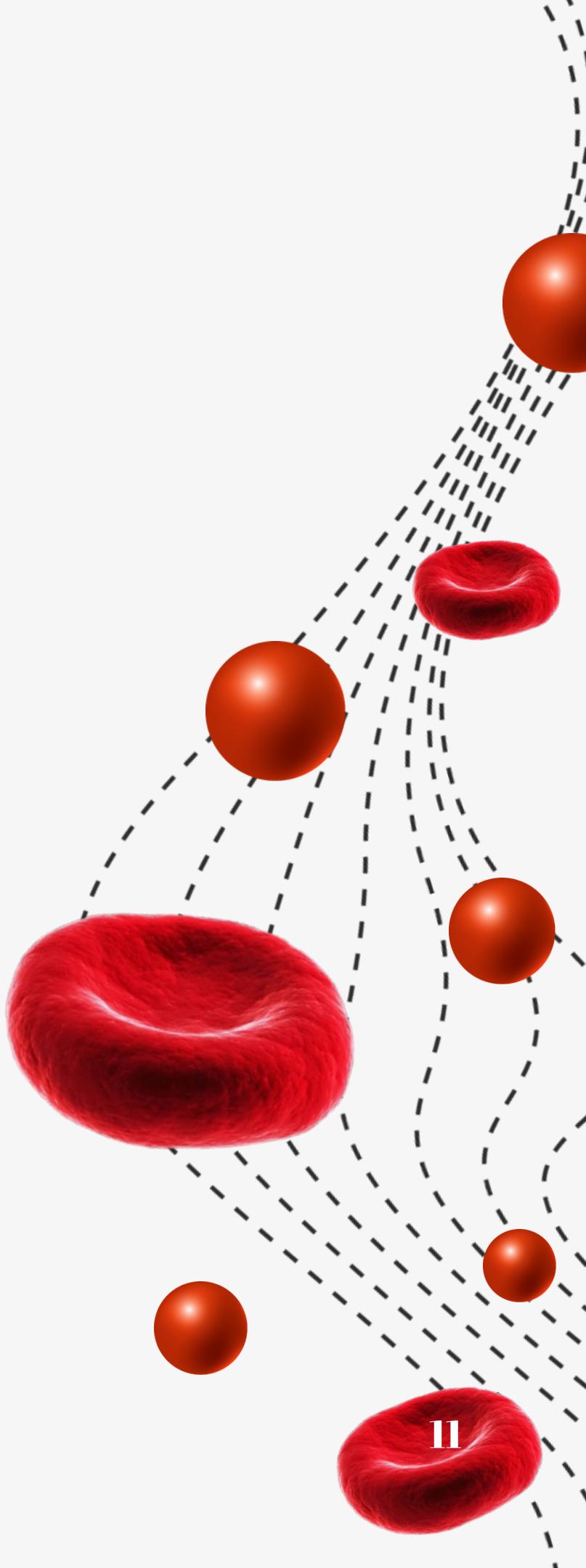


- **Architecture:** Two stacked Inception blocks
- **Features:** 1×1 , 3×3 , 5×5 filters + MaxPool with projection
- **Classifier:** Global average pooling + Dense(64) + Dropout(0.3)
- **Input size:** $128 \times 128 \times 3$
- **Goal:** Capture features at multiple spatial scales

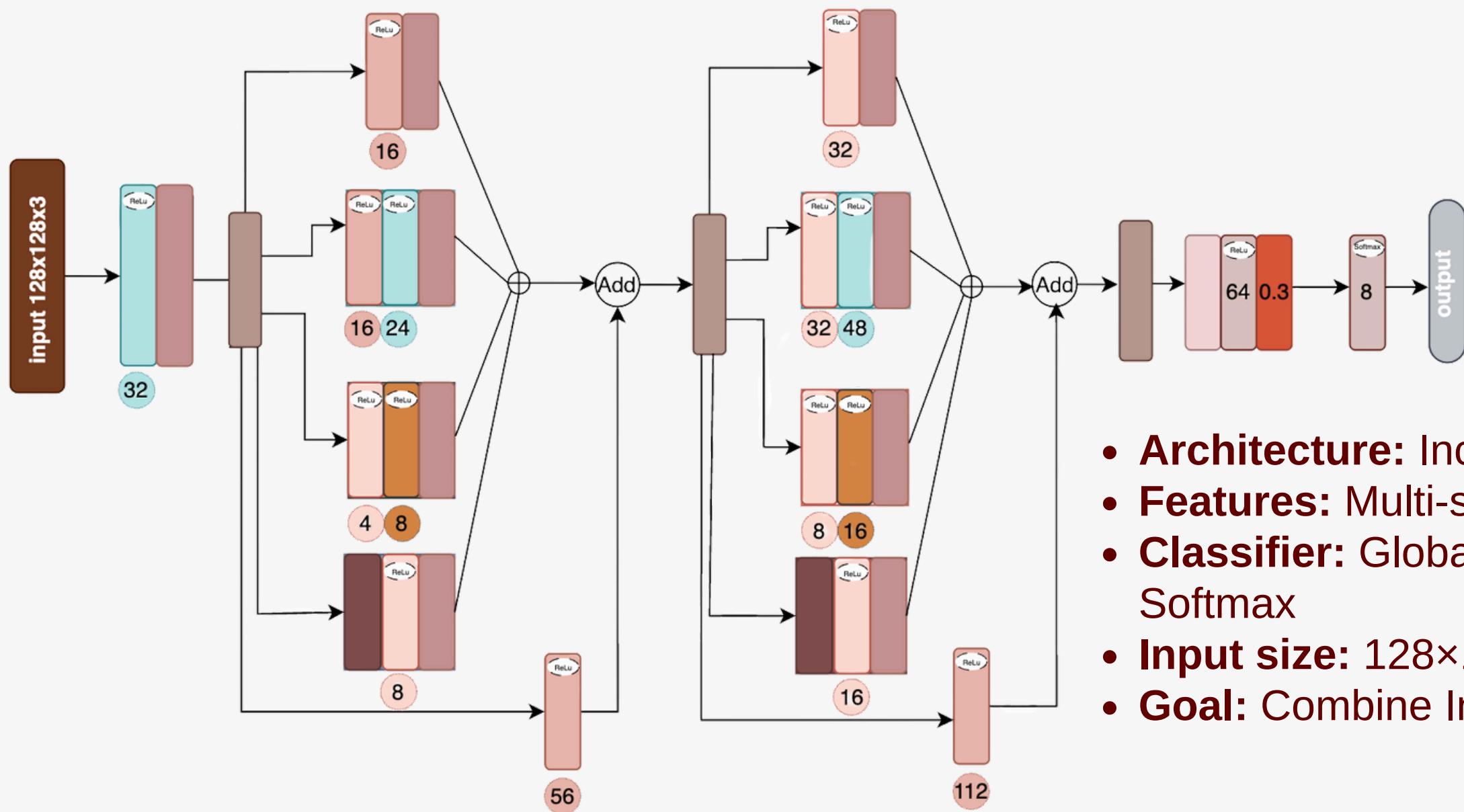
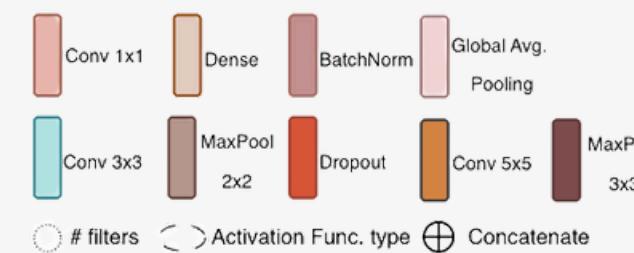
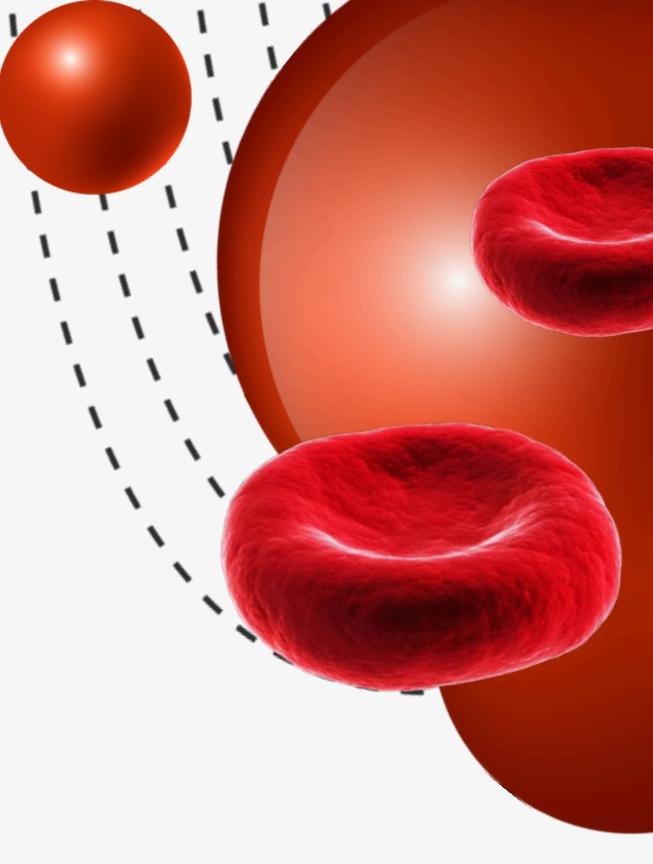
Skip Connection CNN



- **Architecture:** Three residual blocks with skip connections inspired from ResNet
- **Classifier:** Global average pooling + Dense(64) + Dropout(0.3)
- **Regularization:** BatchNorm + Dropout
- **Input size:** $128 \times 128 \times 3$
- **Goal:** Improve training depth and gradient flow



Hybrith (Inception & Skip Connection) CNN



- **Architecture:** Inception blocks with skip/residual connections
- **Features:** Multi-scale convolutions + shortcut connections
- **Classifier:** Global Avg. Pooling → Dense(64) → Dropout(0.3) → Softmax
- **Input size:** $128 \times 128 \times 3$
- **Goal:** Combine Inception's multiscale strength with ResNet stability

Training Configurations

- **Optimizer:** Adam, with a learning rate of 0.0001
- **Loss Function:** Sparse categorical cross-entropy
- **Batch Size:** 64
- **Maximum Epochs:** 40
- **Early stopping** was applied with a patience of 5 epochs, restoring the best model based on validation loss.
- A ReduceLROnPlateau **learning rate scheduler** decreased the learning rate by a factor of 0.5 after 3 stagnant validation epochs.

Results

- All experiments were conducted using a Google Colab Pro environment equipped with an L4 GPU and 22.5 GB of memory.

- **Classification Performance**

- Test accuracy
- Macro-averaged precision, recall, and F1-score
- Class-wise F1-scores (per cell type)

- **Computational Efficiency**

- Training time
- Inference time
- Number of parameters
- Memory usage (MB)

Results – Performance & Complexity

| Model | Parameters (M) | Memory Usage (MB) | Accuracy (%) | Epochs | Train Time (s) | Test Time (s) |
|----------------------------------|----------------|-------------------|--------------|--------|----------------|---------------|
| Baseline CNN (64×64) | 4.29 | 4.36 | 88.66 | 9 | 31.13 | 3.89 |
| Baseline CNN (128×128) | 4.29 | 16.36 | 94.45 | 25 | 154.59 | 4.42 |
| ③ Baseline CNN with Augmentation | 4.29 | 16.36 | 95.88 | 29 | 176.90 | 4.52 |
| ② U-Net Inspired CNN | 2.19 | 8.36 | 96.40 | 25 | 153.26 | 4.38 |
| Inception CNN | 0.04 | 0.14 | 95.03 | 30 | 181.25 | 6.49 |
| ① Skip Connection CNN | 0.31 | 1.17 | 96.78 | 20 | 235.07 | 6.01 |
| Inception & Skip CNN | 0.05 | 0.17 | 94.53 | 32 | 227.08 | 6.73 |

TABLE : Parameter count, estimated memory usage, accuracy, number of epoch and train and test time of CNN architectures.

- **Best Accuracy:** Skip Connection CNN (96.78%), followed closely by U-Net (96.40%) and Baseline with Augmentation (95.88%)
- **Best Efficiency (Low Params & Memory):** Inception CNN (0.04M, 0.14MB) and Inception & Skip CNN (0.05M, 0.17MB)
- **Trade-off Observed:** Some lightweight models (like Inception) have low memory usage but slightly lower performance.
 - Deeper models (like Skip Connection CNN) offer better accuracy with reasonable inference time.
- **Augmentation Boost:** Augmenting the Baseline CNN improved performance from 94.45% to 95.88%, demonstrating regularisation's effectiveness.

Results – Class-Wise Performance Summary

| Model | Precision | Recall | F1-score | Cell Type | Best F1-score Model(s) | Key Observations |
|--------------------------------|-----------|-----------|-----------|----------------|------------------------------------|---|
| Baseline CNN (64×64) | 89 | 87 | 88 | Basophil | U-Net (0.97), Skip (0.93) | U-Net had best F1; Skip had perfect recall but lower precision |
| Baseline CNN (128×128) | 94 | 94 | 94 | Eosinophil | All advanced models (1.00) | Easy to classify; perfect scores with deep/hybrid models |
| Baseline CNN with Augmentation | 96 | 96 | 96 | Erythroblast | Skip, Inception, U-Net (0.98) | Resolution and depth significantly boosted performance |
| U-Net Inspired CNN | 95 | 95 | 95 | Immature Gran. | Skip (0.92), U-Net (0.91) | Skip connection CNN is the best performer |
| Inception CNN | 97 | 97 | 97 | Lymphocyte | Skip (0.99), Inception (0.99) | Deeper models achieved almost perfect results |
| Skip Connection CNN | 97 | 97 | 97 | Monocyte | Skip (0.98), U-Net (0.93) | Skip outperformed all |
| Inception & Skip CNN | 89 | 87 | 88 | Neutrophil | Inception (98.2%), all ≥ 0.95 | All models performed well; even baseline was strong |
| | | | | Platelet | All (1.00 or 0.99) | Easiest class to classify; even baseline achieved near-perfect classification |

Best performer model: Skip Connection CNN

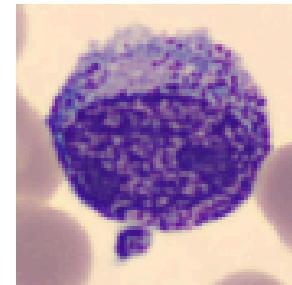


Sample Predictions - Skip Connection CNN

Skip Connection CNN

True: Immature granulocytes

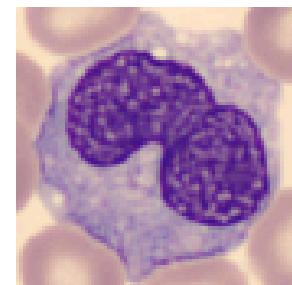
Pred: Immature granulocytes



Skip Connection CNN

True: Monocyte

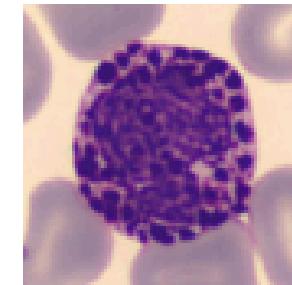
Pred: Monocyte



Skip Connection CNN

True: Basophil

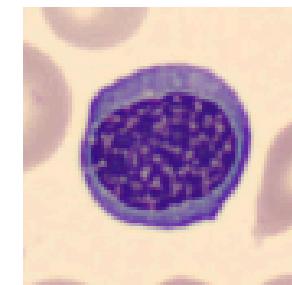
Pred: Basophil



Skip Connection CNN

True: Erythroblast

Pred: Erythroblast



Skip Connection CNN

True: Eosinophil

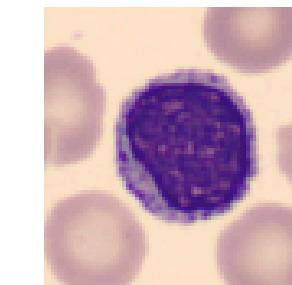
Pred: Eosinophil



Skip Connection CNN

True: Lymphocyte

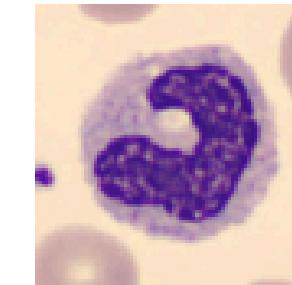
Pred: Lymphocyte



Skip Connection CNN

True: Neutrophil

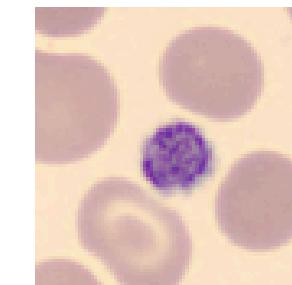
Pred: Immature granulocytes



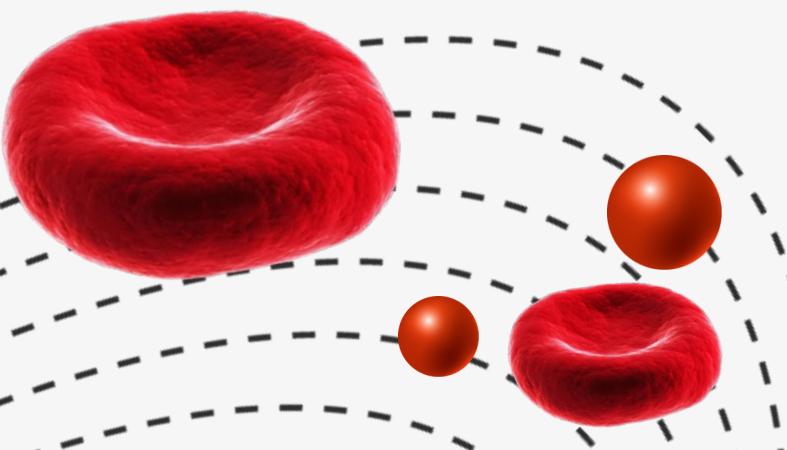
Skip Connection CNN

True: Platelet

Pred: Platelet



Sample test predictions using the Skip Connection CNN model. The true and predicted class labels are shown for each blood cell image. The model correctly classifies nearly all samples, with a single misclassification (Neutrophil predicted as Immature Granulocyte), indicating strong generalisation and robustness across all classes.



Conclusion & Key Takeaways

Key findings

- Baseline CNN highlighted the effects of **resolution** and **augmentation**.
- **U-Net Inspired & Skip Connection CNNs** worked best, especially on less frequent classes.
- **Platelets & Eosinophils**: easiest; **Monocytes & Immature Granulocytes**: hardest.
- Overall, best performer is Skip Connection CNN.

Conclusions

01

02

Advanced CNNs with skip connections or encoder paths are essential for classifying rare or similar blood cells.

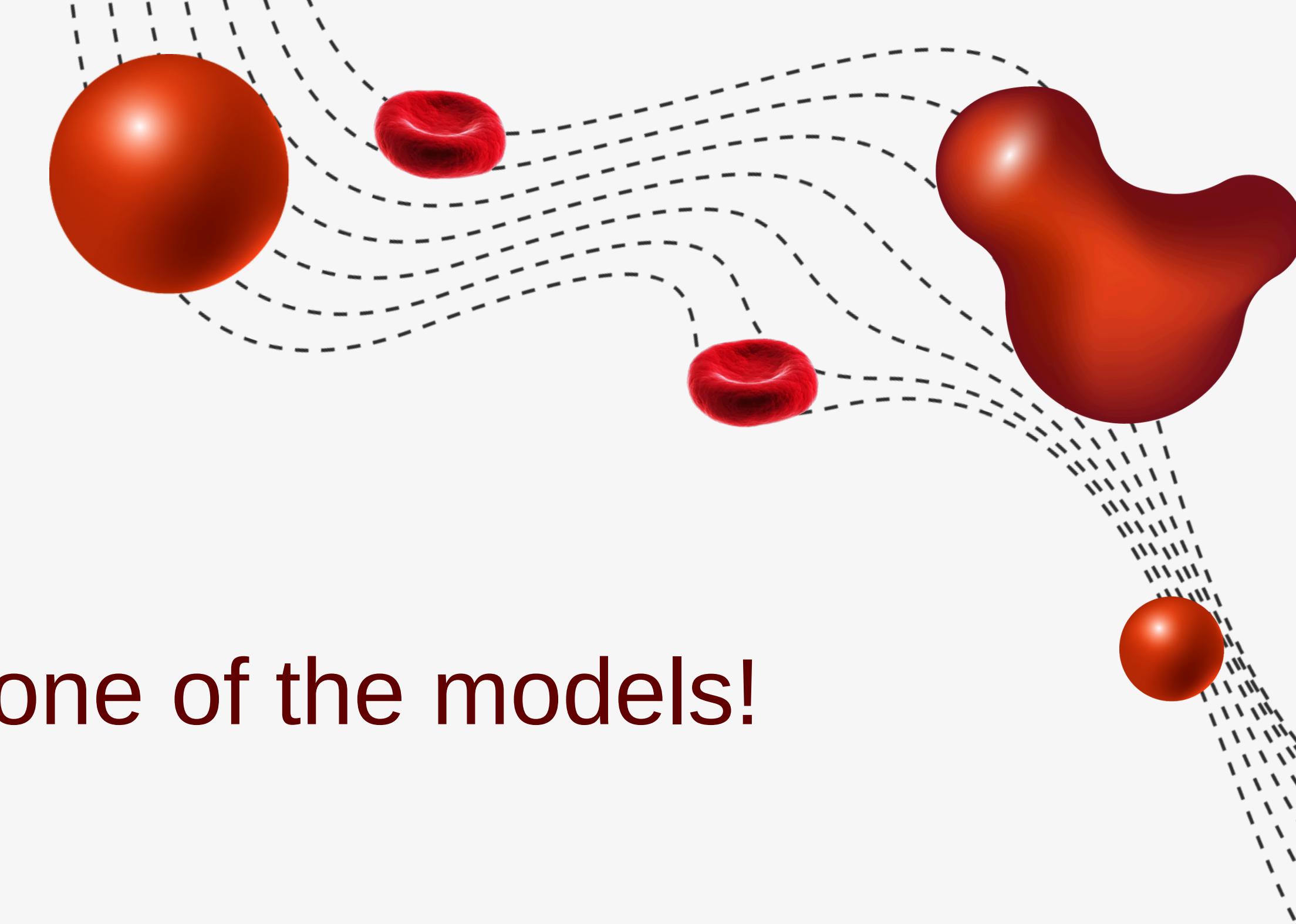
There's a trade-off between depth, accuracy, and efficiency.

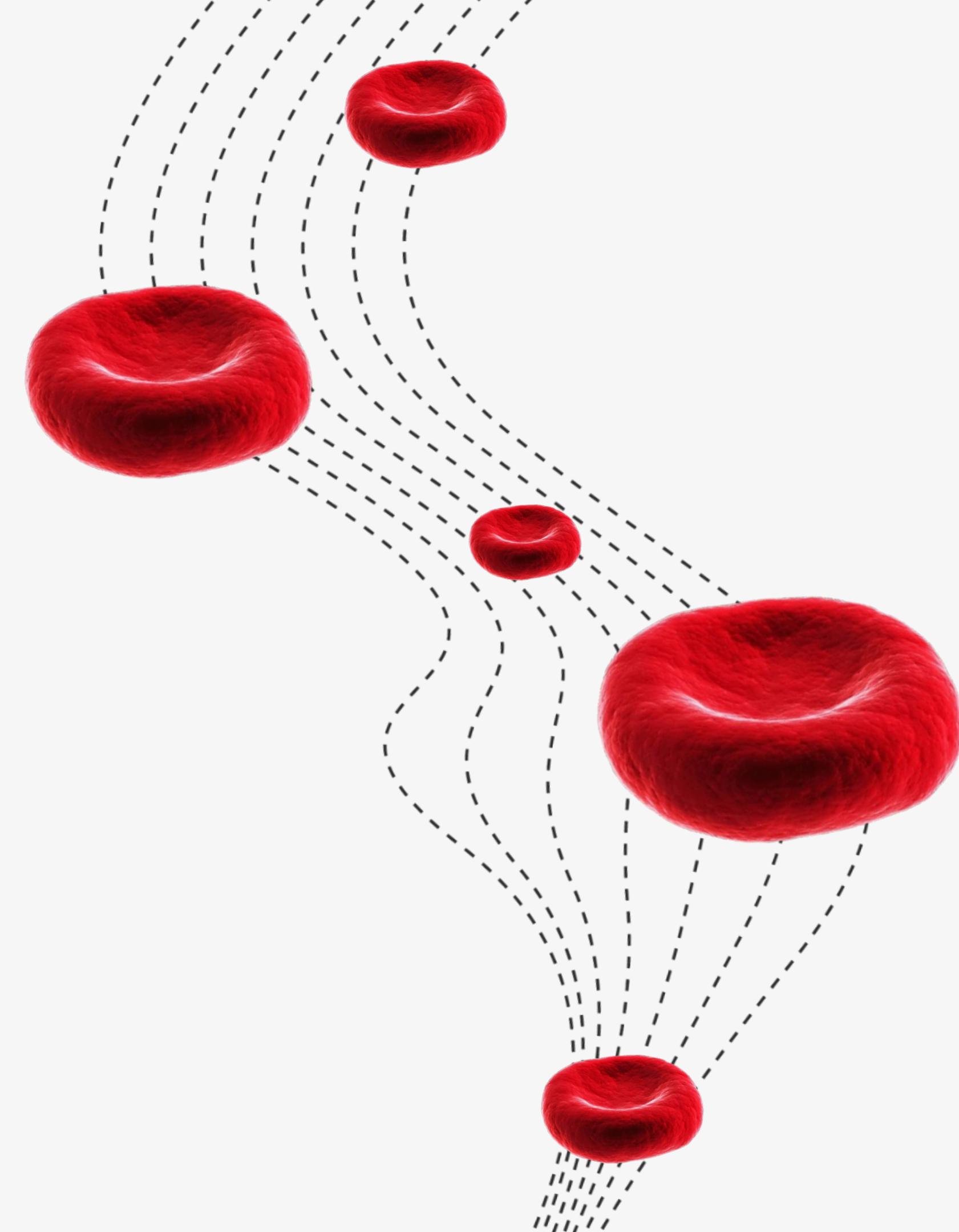
Future Work

- Test models on real clinical datasets for better generalisation.
- Explore attention mechanisms or transformers for harder-to-classify cells.
- Apply targeted data augmentation or class-balanced sampling to boost minority class performance.



Let's try one of the models!





THANKS!

The project can be found on

