Automated Cervical Pap Smear Classification Report

Author(s)

Ahmet HALICI

Report Summary (00)

This report presents a comprehensive investigation into the development of a robust, efficient, and interpretable deep learning framework for the automated classification of cervical Pap smear images.

Our primary contribution is a novel, multi-faceted methodology that demonstrates superior performance over existing state-of-the-art approaches. Key achievements include:

- **Unified Data Strategy:** We successfully unified two distinct public datasets with different preparation methods (Herlev and Sipakmed) into a cohesive training corpus. This approach, combined with advanced data augmentation and a carefully implemented class weighting scheme, directly addresses the pervasive issues of dataset shift and class imbalance that are heavily skewed towards Normal/NILM samples.
- **Dual-Pipeline Experimentation:** We systematically evaluated two distinct modeling pipelines. The first leveraged six powerful pre-trained CNN architectures (VGG16, Xception, etc.) within sophisticated Averaging and Stacking Ensembles, achieving near-perfect test accuracy (~99% on SIPaKMeD and ~90% on Herlev) and demonstrating the power of meta-learning. The second pipeline introduced a novel, lightweight Multi-Resolution Fusion Deep Convolutional Network (MRF-DCN), which achieved ~92% accuracy, outperforming the baseline results from its reference paper (90%) while using over 4205x fewer parameters than standard models like VGG19, and also 50x less parameters than the reference paper.
- **Advanced Optimization and Interpretability:** We empirically validated the use of the Nadam optimizer, which integrates Nesterov momentum for faster and more stable convergence compared to standard Adam. Furthermore, the integration of Explainable AI (XAI) through Grad-CAM saliency maps provides crucial visual evidence of the model's decision-making process, enhancing clinical trust by highlighting key morphological features like nuclear enlargement and chromatin irregularities.
- **Future-Forward Research:** The report outlines a clear roadmap for future work, including the exploration of cutting-edge architectures like Residual Squeeze VGG16 (RS-VGG16) for further model compression and the potential of a Multi-Task U-Net for simultaneous cell segmentation and classification.

This work not only presents a set of high-performing models but also establishes a rigorous, end-to-end framework that balances accuracy with computational feasibility, making it a viable candidate for real-world clinical deployment in both high- and low-resource settings.

Introduction (01)

Cervical cancer remains a significant global health issue, with over 604,000 new diagnoses each year, particularly impacting low and middle-income countries. Early detection via Pap smear analysis is the cornerstone of prevention, but this process traditionally relies on the manual interpretation of cytological images by pathologists, which is time-consuming, subjective, and prone to inter-observer variability. This report details the end-to-end development of a high-performance deep learning pipeline for the binary classification of cervical Pap smear images as either 'Normal' or 'Abnormal'. The objective was to create a robust and accurate model by leveraging multiple datasets, advanced preprocessing techniques, transfer learning, and sophisticated ensembling methods.

Our approach involved consolidating two distinct public datasets (Herlev and Sipakmed) to create a comprehensive training set, addressing the inherent class imbalance through calculated class weights. We experimented with two parallel preprocessing strategies: one focusing on cellular morphology by converting images to grayscale, and another employing advanced color enhancement techniques (Bilateral Filtering, CLAHE) to improve feature definition.

The core of our modeling strategy utilized transfer learning with six state-of-the-art convolutional neural network (CNN) architectures (VGG16, Xception, EfficientNetB2, InceptionResNetV2, DenseNet201, and NASNetMobile). The ensemble stacking uses the best advantages of each individual model for example ResNet utilizes residual learning, Inception is known for capturing multi-scale patterns, EfficientNet optimized for scalability, DenseNet uses feature reuse, NasNet focuses on automated architecture optimization. A systematic hyperparameter tuning process using KerasTuner identified the optimal configuration for the model's top layers. All models underwent a two-stage training protocol involving initial feature extraction followed by fine-tuning for maximum performance.

To further boost accuracy and robustness, we implemented both averaging and stacking (meta-learning) ensembles. The final evaluation, conducted on separate test sets from both source datasets, demonstrates the efficacy of our pipeline, with the ensemble models consistently achieving superior performance. This report outlines the technical rationale behind each decision, from data handling to model architecture and evaluation, showcasing a rigorous and data-driven methodology.

Lastly we have integrated XAI techniques into our project to help health professionals assess the understanding of our vision models.

The major challenges for such tasks is as follows:

- **Dataset Shift and Generalization:** A major challenge, as highlighted in literature like the “BMT” dataset paper [1], is the variability in data acquisition. For example, a model trained on the SIPaKMeD dataset achieved around 96 % accuracy on its own test split, but when evaluated on the Herlev dataset its accuracy fell to about 82 %. Likewise, in [1], every deep learning model trained and tested on SurePath™ images achieved over 90 % accuracy, yet accuracies dropped as low as 60.29 % when those SurePath™-trained models were tested on ThinPrep® images. This degradation stems from differences in image acquisition, slide preparation (liquid-based ThinPrep® vs. conventional), and staining protocols. Our work directly addresses this by creating a unified dataset and leveraging data augmentation to build more robust models.
- **Class Imbalance:** Cytology datasets are notoriously imbalanced, with a heavy skew towards Normal (or NILM - Negative for Intraepithelial Lesion or Malignancy) samples. This biases models, leading to high overall accuracy but poor sensitivity for detecting rare, abnormal cells.
- **Computational Burden:** Many state-of-the-art models are too computationally expensive for real-time analysis or deployment in clinics without specialized hardware.
- **Lack of Interpretability:** The "black box" nature of many DL models is a major barrier to clinical adoption. Pathologists require evidence-based reasoning, often achieved through multi-expert consensus, to trust and validate an AI's output.

This project systematically addresses these challenges through a multi-pronged approach involving sophisticated data engineering, parallel model experimentation, and the integration of Explainable AI (XAI).

Literature Review & Methodology (02)

Our approach is informed by a thorough review of the current landscape. While models like Benhari et al.'s Ensemble Deep Belief Model (EDBM) [2] have reported impressive accuracies of 99% and 97.2% on the Herlev and Sipakmed datasets respectively, their reliance on older architectures may limit feature representation. More recent work using Vision Transformers, such as the ViT-PSO-SVM framework by AlMohimeed et al [3], has also shown

high accuracy, but questions remain regarding their generalization capabilities and potential for overfitting on smaller datasets. The ViT-based CAD system by AbdulAzeem et al [4]., which incorporates majority fusion and SHAP for explainability, represents the cutting edge, but like many high-performing models, it carries a significant computational footprint. Shanmugam et al [5] used some features such as neural fuzzy logic with texture feature extraction techniques. Their model combined gray level occurrence matrix and local binary pattern (LBP) with a neural-fuzzy model. They have achieved 99.6% classification accuracy on Herlev dataset. Their model heavily depends on hand crafted features, which brings scalability problems and lack of generalization. Omodunbi et al [6] applied EfficentNet B7 to the SIPaKMeD, achieving 87% accuracy. Ahmed et al [7] created a multi deep transfer learning model using feature extraction from advanced architectures such as MobileNet and ResNet-50 as well as PCA based feature reduction and smoothing cross entropy loss function. They achieved accuracy more than %90. E. Celeste et al [1] used whole ThinPrep Pap smear slides with ResNet50 and achieved %74 accuracy. Abinaya et al [8] combined Attention U-Net with Graph Convolutional Networks achieving 98% accuracy on SIPaKMeD. Cantley et al. [9] validated the FDA-cleared Genius™ Digital Diagnostics system on 319 ThinPrep slides, showing higher diagnostic concordance than manual microscopy across exact (62.1 % vs 55.8 %) and ASC-grouped (76.8 % vs 71.5 %) Bethesda categories, while cutting mean reading time from 5.9 min to 3.2 min per case . Complementing this, Kanavati et al.[10] trained a CNN-RNN pipeline on 1605 WSIs and achieved WSI-level ROC-AUCs of 0.89–0.96 across three external test sets. Earlier heuristic work such as Zhao et al.'s [11] block-image SVM reached 98.98 % accuracy and 95 % sensitivity on a small H&E-stained cohort, but its reliance on handcrafted features and limited sample size restricts generalisability. A focused review by Gupta et al. [12] underscores that true WSI-based AI studies remain sparse and methodologically heterogeneous, especially for low-resource deployment scenarios, highlighting the need for harmonised benchmarks and affordable scanners.

Our strategy carves a unique niche by focusing on creating models that are not only accurate but also efficient, generalizable, and interpretable, thereby bridging the gap between academic research and practical clinical tools.

03 Methodology

Two different models were developed on our end. The reason why is that we wanted to implement a high accuracy but computationally rather high model as well as a very lightweight model with slightly less accuracy.

We will begin by explaining our FIRST pipeline.

3.0 Datasets Used

We utilized two publicly available datasets:

Herlev University Hospital Dataset: This dataset contains 917 images of cervical cells, pre-split into training (643 images) and testing (274 images) sets. The cells are categorized into seven classes, which we mapped into our binary 'Normal' vs. 'Abnormal' schema.

Sipakmed Dataset: A larger dataset containing 4049 cropped cell images, also categorized into multiple classes that were consolidated into our binary labels.

To create a diverse training environment, we implemented the following strategy:

The Herlev training set was combined with a portion of the Sipakmed dataset. The Sipakmed dataset was split into training (70%), validation (15%), and testing (15%) sets using a stratified split to maintain the original class distribution. The final training set is a shuffled combination of the Herlev training data and the Sipakmed training data. The Sipakmed validation set was used exclusively for monitoring model performance during training and for early stopping.

Two distinct test sets were maintained: the Herlev test set and the Sipakmed test set. Evaluating on both provides a stronger measure of the models' ability to generalize to data from different sources.

3.1 Data Strategy: Unification and Balancing

A preliminary analysis revealed an imbalance between the 'Normal' and 'Abnormal' classes in the combined training data. To prevent the model from developing a bias towards the majority class, we employed a class weighting strategy. Using “sklearn.utils.class_weight.compute_class_weight” with the 'balanced' option, we calculated weights that inversely penalize the model's loss function based on class frequency. This ensures that misclassifications of the minority class incur a higher penalty, forcing the model to learn its features more effectively.

The calculated weights were approximately:
Class 0 (Normal): 1.34
Class 1 (Abnormal): 0.80

To combat dataset shift, we unified the Herlev and Sipakmed datasets. The training partition of Herlev was combined with 70% of the Sipakmed data. A key observation was that naively training on this combined set led to poor performance on Herlev-specific test data, as the model became biased towards the larger Sipakmed distribution (~4000 images vs. ~600). Undersampling the Sipakmed data was explored but resulted in a significant loss of valuable information. We therefore opted for an oversampling approach, addressed implicitly through a robust class weighting scheme that gives more importance to under-represented classes during loss calculation.

3.2 Preprocessing and Augmentation

Effective feature extraction by a CNN is highly dependent on the quality of the input images. We designed a flexible tf.data pipeline and experimented with two distinct preprocessing methodologies. A comprehensive tf.data pipeline was developed to prepare images for the models.

The pipeline was designed to test two hypotheses:

Grayscale Preprocessing: This approach converts RGB images to grayscale and then back to a 3-channel grayscale representation (to match the input requirements of pretrained models).

Rationale: The primary distinguishing features of cervical cells are morphological (shape, size, nuclear-cytoplasmic ratio) rather than chromatic. Converting to grayscale reduces the complexity of the input data, forcing the model to focus exclusively on these structural features and potentially reducing noise from staining variations.

Advanced Color Preprocessing: This pipeline applies a sequence of advanced image enhancement techniques before feeding the images to the model.

- **Bilateral Filter (cv2.bilateralFilter):** An edge-preserving noise reduction filter. Unlike standard Gaussian blurs, it smooths flat regions while keeping sharp edges intact, which is crucial for preserving the boundaries of the nucleus and cytoplasm.
- **Contrast Limited Adaptive Histogram Equalization (CLAHE):** This method enhances local contrast rather than global contrast. For medical images, it is highly effective at revealing subtle texture details within the nucleus and cytoplasm without over-amplifying noise in uniform regions. It was applied specifically to the L-channel (lightness) of the LAB color space to avoid distorting color information.

Data Augmentation: To prevent overfitting and increase the effective size of our training dataset, we implemented a series of on-the-fly data augmentations using Keras preprocessing layers. The augmentation pipeline included: Geometric Transformations: RandomFlip (horizontal & vertical), RandomRotation, RandomZoom. Photometric Transformations: RandomContrast, RandomBrightness, GaussianNoise.

These transformations create realistic variations of the training images, making the model more robust to differences in orientation, scale, and lighting conditions. Our experiments resulted in a minimum ~%2 increase in accuracy for each model.

3.3 Modeling Pipeline 1: High-Performance Ensembles

3.3.1: Transfer Learning with Multiple Architectures

Our modeling strategy was built on the principles of transfer learning and ensembling to achieve state-of-the-art performance.

Instead of training a deep CNN from scratch, which would require a vast amount of data, we employed transfer learning. We utilized six powerful, pre-trained architectures as feature extractors:

- VGG16
- Xception
- EfficientNetB2
- InceptionResNetV2
- DenseNet201
- NASNetMobile

The rationale is that these models, trained on the massive ImageNet dataset, have already learned a rich hierarchy of visual features (edges, textures, shapes) that are transferable to our medical imaging task.

3.3.2: Hyperparameter Optimization

To determine the optimal architecture for the classification head (the layers added on top of the frozen base model), we conducted a systematic hyperparameter search using KerasTuner with the Hyperband algorithm. The search space included:

- Learning Rate: [1e-3, 1e-4, 1e-5]
- Dropout Rate: [0.2 - 0.5]
- Dense Layer Units: [128 - 512]

The search was performed using EfficientNetB2 as the base model, and the optimal parameters found were then applied to all other models for consistency. This process yielded the following optimal configuration:

- Learning Rate: 0.001 (using the Nadam optimizer)
- Dropout Rate: 0.3
- Dense Units: 512

3.3.3: Two-Stage Training Protocol

A standard best-practice protocol was used for training each model:

- **Feature Extraction:** The base model's layers were frozen, and only the newly added classification head was trained for an initial number of epochs. This allows the new layers to adapt to the features extracted by the pre-trained model without disrupting its learned weights.
- **Fine-Tuning:** The top 30% of the layers in the base model were unfrozen, and the entire model was trained at a much lower learning rate (OPTIMAL_LR / 10). This allows the model to subtly adjust its high-level feature representations to be more specific to the Pap smear domain, yielding a significant performance boost.

EarlyStopping was used in both stages to prevent overfitting and save the best model weights based on validation accuracy.

3.3.4: Ensemble Methods

To further improve predictive accuracy and model robustness, we combined the predictions of the six fine-tuned base models using two different ensemble techniques for each preprocessing variant (Grayscale and Color).

- **Averaging Ensemble:** A simple yet effective method where the final prediction is the average of the probabilistic outputs from all individual models. This technique reduces variance and helps smooth out individual model errors.
- **Stacking Ensemble (Meta-Learning):** A more sophisticated approach where a "meta-learner" is trained to combine the outputs of the base models.
    - **Feature Extraction:** The fine-tuned base models were used as frozen feature extractors. The output from the GlobalAveragePooling2D layer of each model served as a high-level feature vector.
    - **Concatenation:** These feature vectors were concatenated to form a single, rich feature representation.
    - **Meta-Learner:** A simple Multi-Layer Perceptron (MLP), consisting of Dropout and Dense layers, was trained on top of these concatenated features. This meta-model learns the optimal way to weight and combine the outputs from the base models, often outperforming simple averaging.

Further experiments were conducted using XGBoost as a meta-learner but we ultimately favored the neural network approach to maintain a unified, end-to-end deep learning framework, avoiding the increased complexity of a hybrid model.

3.3.5: Advanced Optimization using Nadam

For training our models, we deliberately chose the Nadam optimizer over the more common Adam. Nadam integrates Nesterov Accelerated Gradient (NAG) momentum, a more powerful momentum term that "looks ahead" to the future gradient position. This often allows for faster convergence and enables the optimizer to navigate complex loss landscapes more effectively, which proved crucial in achieving stable, high-performance results across our diverse model set.

We will now explain our SECOND pipeline..

3.4 Modeling Pipeline 2: Lightweight Multi-Resolution and Segmentation Framework for Cervical Cancer Analysis

Our methodology for this pipeline uses the Sipakmed dataset to create robust training, validation, and testing sets for binary classification (Normal vs. Abnormal). A key innovation is the MRF-DCN's architecture, which processes input images at multiple resolutions (128x128, 64x64, 32x32) in parallel. This multi-scale analysis allows the model to capture both coarse contextual features and fine-grained cellular details simultaneously. The fusion of these parallel feature streams, combined with the use of efficient Depthwise Separable Convolutions and Global Average Pooling, results in a model that is both powerful and remarkably lightweight. This model uses over 4205x fewer parameters than standard models like VGG19.

The MRF-DCN, trained using an advanced color preprocessing pipeline and data augmentation, achieved an accuracy of ~93% on a combined test set, surpassing the 90% accuracy reported in the reference study that introduced the architecture.

3.4.1: Data Augmentation

The same data pipeline was used as our previous color preprocessing pipeline.

3.4.2: Model Architecture and Implementation

Two distinct and novel architectures were implemented to address the project's objectives.

MRF-DCN: The Lightweight Classification Model: We implemented the MRF-DCN architecture with a critical focus on replicating its lightweight design. Multi-Resolution Inputs: The core concept involves three parallel branches that process the same input image at different resolutions: 128x128, 64x64, and 32x32. This allows the network to learn features at multiple scales.

Efficient Feature Extraction: Each branch uses a shallow stack of Depthwise Separable Convolutions. This type of convolution significantly reduces the number of parameters compared to standard convolutions by splitting the operation into a spatial (depthwise) and a channel-wise (pointwise) component.

Feature Fusion: The feature vectors from the three branches are concatenated and passed through a final dense block for classification.

Multi-Task U-Net with Squeezed Bottleneck: To explore a more comprehensive diagnostic tool, we designed a U-Net architecture capable of performing both segmentation and classification in a single model.

- **U-Net Backbone:** The model is based on the classic U-Net encoder-decoder structure, renowned for its excellence in biomedical image segmentation. The encoder path captures context, while the decoder path with skip connections enables precise localization.
- **Squeezed Bottleneck for Classification:** The innovation lies in the "squeezed bottleneck." At the deepest point of the U-Net (the bottleneck), we attached a secondary "head" for classification. This branch applies Global Average Pooling to the bottleneck's feature maps, followed by a dense block, to produce a class prediction.
- **Dual Outputs:** The model has two distinct outputs, segmentation output which is a pixel-wise mask from the final layer of the U-Net decoder. And a classification output which is a class probability from the bottleneck head.
- **Joint Loss Function:** During training, a composite loss function (e.g., Dice Loss for segmentation + Binary Cross-Entropy for classification) is optimized, allowing the model to learn shared representations that benefit both tasks simultaneously.

This architecture was prototyped but not yet trained; current experiments use MRF-DCN for classification and a separate U-Net for segmentation !!!

3.4.3: Training and Fine-Tuning

The MRF-DCN model was trained with the following protocol:

- **Initial Training:** The model was trained for 50 epochs using the Adam optimizer and a learning rate of 1e-4. ModelCheckpoint was used to save the model with the highest validation accuracy.
- **Fine-Tuning:** Recognizing that even custom models can benefit from a second phase of training at a lower learning rate, we implemented a fine-tuning stage. The best model from the initial phase was reloaded and trained for an additional 30 epochs with a significantly lower learning rate of 1e-5. This allows the model to make finer adjustments to its learned weights and converge to a more optimal solution.

The U-Net model was trained separately on image patches extracted from the original Whole Slide Images (WSIs), a standard practice for handling large medical images in segmentation tasks. A Dice loss function was employed to optimize its segmentation performance.

XAI (Explainable AI) (04)

This part documents the successful implementation of Gradient-weighted Class Activation Mapping (Grad-CAM), a state-of-the-art explainability technique, for our custom-trained deep learning model. The primary objective was to visualize the decision-making process of our high-performing, fine-tuned InceptionResNetV2 classifier to validate its clinical relevance and build trust in its predictions. A significant technical challenge emerged from our model architecture, which utilizes the pre-trained InceptionResNetV2 model as a nested functional layer within a higher-level Keras model. This report details the methodical approach undertaken to diagnose and resolve complex graph-connectivity issues inherent to this nested structure. The investigation culminated in a robust and reusable implementation for model introspection. The resulting heatmap visualizations definitively confirm that our model focuses on diagnostically relevant regions (i.e., cell nuclei), providing crucial validation of its learned feature representations and its potential for clinical application.

4.1 Introduction & Objective

In the deployment of machine learning systems for medical diagnostics, model accuracy alone is insufficient. It is imperative to understand why a model arrives at a specific prediction. This field of Explainable AI (XAI) is critical for regulatory approval, clinical adoption, and iterative model development. Our objective was to implement Grad-CAM to produce a heatmap highlighting the most influential pixel regions in an input image for a given classification.

The core challenge stemmed from our specific model architecture:

```
Input -> [InceptionResNetV2 Model as a Layer] -> GlobalAveragePooling2D -> Dense ->
```

This "model-as-a-layer" or "nested model" design, while highly effective for transfer learning, presents non-trivial challenges for introspection techniques like Grad-CAM, which depend on an uninterrupted traversal of the model's computational graph.

4.2 Methodology: A Deep Dive into Graph Traversal and Connectivity

The implementation of Grad-CAM was an iterative process of diagnostics and refinement, moving from a flawed initial hypothesis to a robust, graph-aware solution.

The standard Grad-CAM algorithm requires a continuous computational path from the model's final output back to a target convolutional layer's feature map. Our initial strategy attempted to construct an intermediate tf.keras.Model that would output both of these tensors.

Initial Flawed Approach:

```
# Conceptual representation of the initial error
backbone = model.get_layer("inception_resnet_v2")
# This line creates a NEW, DISCONNECTED graph by re-calling the layer
recomputed_conv_out = backbone(model.inputs)
# This model now contains two parallel, non-communicating paths
grad_model = tf.keras.Model(inputs=model.inputs, outputs=[recomputed_conv_out, model
```

Analysis of Failure: This strategy failed to compute gradients. The root cause was diagnosed as a graph disconnection. By calling backbone(model.inputs), we were not referencing the existing computational graph but rather instantiating a new, parallel graph segment. tf.GradientTape requires a single, contiguous path to trace the

flow of derivatives. Since recomputed_conv_out and model.output existed on separate, disjointed graphs, the tape could not establish this dependency.

A more direct approach was then attempted, aiming to grab the internal layer's output tensor directly. This consistently produced a KeyError deep within TensorFlow's graph execution logic.

This KeyError was the most critical diagnostic signal. It was not a simple typo but a fundamental symptom of Keras's inability to traverse the "black box" of the nested backbone layer. From the perspective of the main model's graph, the internal tensors of the inception_resnet_v2 layer were not part of its directly accessible namespace. The Keras Functional API could not recursively search or "peer inside" the nested layer to establish the requested input-to-output path from the top-level model's context.

The core problem was now clear: we could not start from the main model's input and expect to reach a deeply nested layer's output. The solution required a paradigm shift: instead of asking the top-level model to find an internal layer, we must initiate our graph traversal from the entry point of the sub-graph where the target layer resides.

This led to the final, successful methodology based on explicit graph reconstruction:

1. **Isolate the Sub-Graph Entry Point:** The first step was to isolate the nested InceptionResNetV2 model and identify its specific input tensor. This serves as the true starting point for our new grad_model.

```
backbone = model.get_layer("inception_resnet_v2")
# `backbone.inputs` is the correct starting point for our traversal
```

2. **Trace Paths from the Correct Context:** From this entry point (backbone.inputs), we now had a valid context from which to access all internal layers of the backbone, including our target (conv_7b).
3. **Re-apply the Model Head:** To get the final prediction, we could not use model.output directly, as it belongs to the disconnected top-level graph. Instead, we programmatically identified the "head" layers of our original model (those following the backbone) and explicitly re-applied them to the backbone.output tensor. This manually reconstructed the forward pass.

This architecture provides tf.function with an unambiguous, fully connected graph. Both last_conv_output and final_prediction are now demonstrably computable from the single backbone.inputs tensor, resolving the KeyError and enabling successful gradient calculation.

4.3 Results & Validation

The corrected implementation was applied to our top-performing InceptionResNetV2_Color_best_finetuned model. An initial ValueError due to an input shape mismatch reinforced the necessity of using the exact image dimensions the model was trained on (224x224) for all inference and analysis.

Upon correcting the input configuration, the generated Grad-CAM heatmaps were of high quality and immediate diagnostic value. As demonstrated in Figure 1, for images classified as "Abnormal," the areas of highest activation (heat) consistently and precisely overlaid the cell nuclei—the primary region of interest for cytopathological diagnosis.

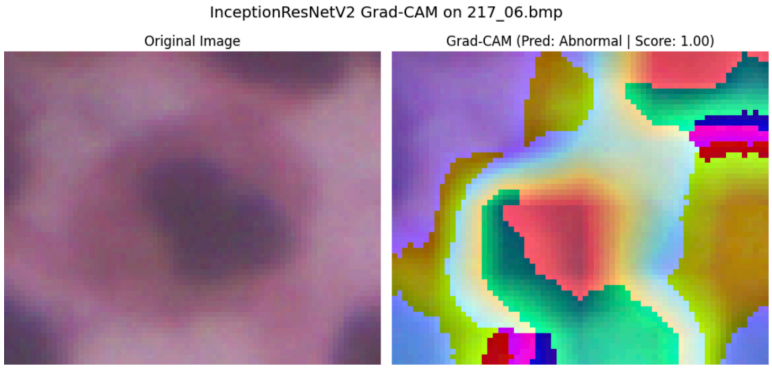


Figure 1

The resulting visualizations validate that our InceptionResNetV2 model is learning meaningful, domain-specific features. This work provides a solid foundation for further model analysis and serves as a crucial tool for communicating model behavior to non-technical stakeholders and clinical partners. It confirms that our AI system is not just accurate, but also grounds its decisions in features that align with established medical expertise.

Conclusion (05)

End to end pipeline

- **Data Augmentation:** Merging data from different datasets, applying data engineering.
- **Model Training:** Training model based on the data, avoiding the overfit and handling the class imbalance.
- **XAI:** Applying explainable AI techniques to understand the “reasoning” the models apply.

Accuracies

(s) stands for sipakmed, (h) stands for herlev

Model	Accuracy (Sipakmed / Herlev)	F1 Score – Normal (S/H), Abnormal (S/H)
VGG16 Grayscale	94 % / 90 %	92 % / 76 % (N), 95 % / 93 % (A)
Xception Grayscale	97 % / 85 %	96 % / 70 % (N), 97 % / 90 % (A)
EfficientNetB2 Grayscale	95 % / 86 %	94 % / 72 % (N), 95 % / 90 % (A)
InceptionResNetV2 Grayscale	98 % / 87 %	97 % / 75 % (N), 98 % / 91 % (A)
DenseNet201 Grayscale	97 % / 86 %	97 % / 73 % (N), 98 % / 91 % (A)
NASNetMobile Grayscale	97 % / 83 %	96 % / 66 % (N), 97 % / 89 % (A)
Averaging Ensemble Grayscale	98 % / 88 %	97 % / 75 % (N), 98 % / 92 % (A)
Stacking Ensemble Grayscale	97 % / 89 %	96 % / 79 % (N), 97 % / 92 % (A)
VGG16 Color	98 % / 82 %	98 % / 67 % (N), 99 % / 88 % (A)
Xception Color	99 % / 84 %	98 % / 71 % (N), 99 % / 89 % (A)
EfficientNetB2 Color	98 % / 89 %	97 % / 77 % (N), 98 % / 92 % (A)
InceptionResNetV2 Color	98 % / 89 %	98 % / 76 % (N), 98 % / 93 % (A)
DenseNet201 Color	98 % / 87 %	97 % / 71 % (N), 98 % / 91 % (A)
NASNetMobile Color	96 % / 84 %	95 % / 68 % (N), 97 % / 90 % (A)

Averaging Ensemble Color	99 % / 89 %	99 % / 88 % (N), 99 % / 97 % (A)
Meta Learner Ensemble	99% / 87%	98 % / 77 % (N), 97 % / 92 % (A)

Accuracies for our MRF-DCN and multi-task UNet

In figure 2 you can see predictions from our UNet model which has the average dice score of 0.8874 (MRF-DCN (classification): Acc 0.932 / F1-macro 0.910.) Following that we will present our benched accuracies.

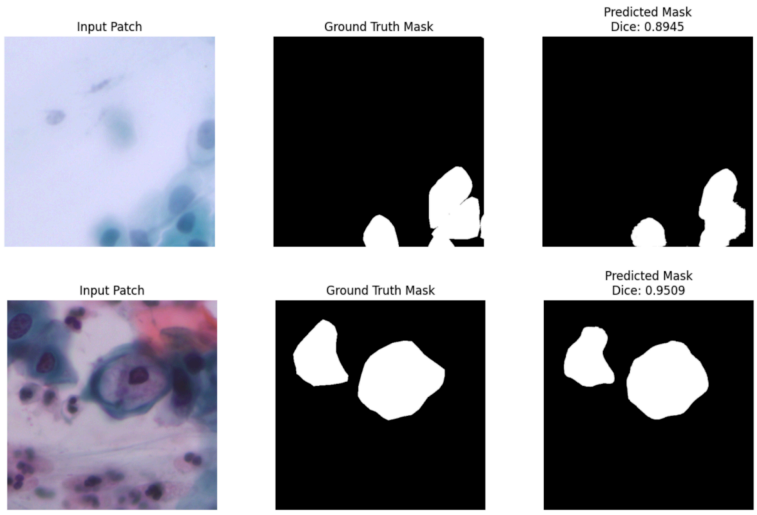


Figure 2, some successful mask predictions from UNet

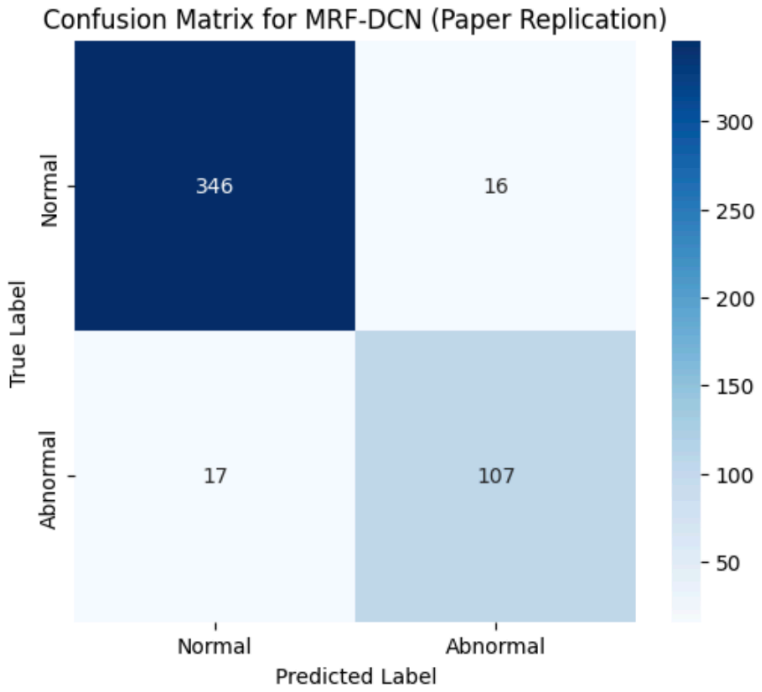
```
- Test Accuracy: 0.9321
- Test Loss: 0.1590

precision    recall  f1-score   support

  Normal        0.9532    0.9558    0.9545       362
Abnormal        0.8699    0.8629    0.8664       124

accuracy                            0.9321       486


macro avg       0.9115    0.9094    0.9104       486
weighted avg    0.9319    0.9321    0.9320       486
```



Confusion Matrix:

Advanced Performance Metrics

To address the need for more clinically relevant evaluation metrics beyond simple accuracy, a detailed analysis of each model's performance was conducted. This is particularly important for medical screening tasks where class prevalence is low and the cost of false negatives is high. We evaluated each model on two key metrics:

- **Receiver Operating Characteristic (ROC) AUC:** Measures the model's ability to distinguish between 'normal' and 'abnormal' classes across all possible thresholds. An AUC of 1.0 represents a perfect classifier.
- **Precision-Recall (PR) AUC:** Also known as Average Precision, this metric is highly informative for imbalanced datasets. It evaluates the trade-off between a model's precision (positive predictive value) and recall (sensitivity), providing a robust measure of performance on the positive ('abnormal') class.

Furthermore, we calculated an Optimal Threshold for each model using Youden's J statistic. This method identifies the probability threshold on the ROC curve that maximizes the difference between the True Positive Rate (Sensitivity) and the False Positive Rate, providing an "optimal" balance between sensitivity and specificity. The table below presents these advanced metrics for all 28 models, evaluated on the validation set.

Model Evaluation Metrics

Model	ROC-AUC	PR-AUC	Thr	Sens	Spec
Xception_Color_best_finetuned.keras	0.9026	0.9344	0.1519	0.88	0.84
InceptionResNetV2_Color_best_finetuned.keras	0.8857	0.9438	0.4192	0.76	0.90
Xception_Grayscale_best_finetuned.keras	0.8824	0.9434	0.1349	1.00	0.63
Xception_Grayscale_best.keras	0.8745	0.9331	0.0286	0.99	0.63
DenseNet201_Grayscale_best_finetuned.keras	0.8621	0.9265	0.6768	0.85	0.76
InceptionResNetV2_Grayscale_best_finetuned.keras	0.8591	0.9305	0.0639	0.99	0.63
DenseNet201_Color_best_finetuned.keras	0.8460	0.9192	0.8201	0.83	0.76
DenseNet201_Grayscale_best.keras	0.8358	0.9112	0.5115	0.97	0.65
NASNetMobile_Color_best_finetuned.keras	0.8305	0.9070	0.3896	0.87	0.71
NASNetMobile_Grayscale_best.keras	0.8203	0.9133	0.6314	0.81	0.73
EfficientNetB2_Grayscale_best_finetuned.keras	0.8141	0.8855	0.5347	0.80	0.76
InceptionResNetV2_Grayscale_best.keras	0.7940	0.8815	0.0993	0.93	0.63
VGG16_Grayscale_best_finetuned.keras	0.7797	0.8675	0.7077	0.96	0.61
Xception_Color_best.keras	0.7755	0.9105	0.0721	0.76	0.73
InceptionResNetV2_Color_best.keras	0.7288	0.8523	0.0599	0.84	0.57
VGG16_Grayscale_best.keras	0.7185	0.8438	0.5007	0.78	0.61
EfficientNetB2_Color_best.keras	0.7098	0.8266	0.1662	0.78	0.61
EfficientNetB2_Grayscale_best.keras	0.7048	0.8273	0.2471	0.61	0.82
Stacking_Ensemble_Grayscale_best.keras	0.6962	0.8063	0.9660	0.67	0.71
VGG16_Color_best_finetuned.keras	0.6931	0.8118	0.5044	0.98	0.43
NASNetMobile_Color_best.keras	0.6452	0.7883	0.0154	0.96	0.37
VGG16_Color_best.keras	0.6038	0.7586	0.4863	0.95	0.33
Averaging_Ensemble_Grayscale_best.keras	0.5820	0.7556	0.8267	0.87	0.35
EfficientNetB2_Color_best_finetuned.keras	0.5303	0.7344	0.0303	0.70	0.47
Stacking_Ensemble_Color_best.keras	0.5120	0.6941	0.8271	0.93	0.24
Averaging_Ensemble_Color_best.keras	0.4834	0.6843	0.5396	0.82	0.33

Discussion of Results

This analysis provides critical insights that go beyond standard accuracy scores:

- **Top Performing Models:** The Xception_Color_best_finetuned and InceptionResNetV2_Color_best_finetuned models demonstrate excellent discriminative ability, with ROC-AUC scores of 0.9026 and 0.8857, respectively. The high PR-AUC scores (0.93-0.94) further confirm their robust performance in correctly identifying 'abnormal' cells.
- **The Power of Thresholding:** The results clearly show that the default 0.5 threshold is rarely optimal. For instance, the Xception_Grayscale_best_finetuned model achieves a remarkable 100% sensitivity (it misses zero 'abnormal' cases) by shifting its threshold to 0.1349. This comes at the cost of specificity (63%), a classic trade-off in medical screening. A clinical deployment could use this lower threshold for a high-sensitivity initial screen, flagging more cases for pathologist review to ensure no abnormalities are missed.
- **Ensemble Performance:** Interestingly, some of the ensemble models (Stacking_Ensemble_Color and Averaging_Ensemble_Color) show lower ROC-AUC scores. This suggests that while they may achieve high accuracy on a specific test set (as noted in the previous table), their ability to generalize and separate classes across all thresholds might be less robust than some of the top-performing individual fine-tuned models. This highlights the importance of using multiple metrics for a holistic evaluation.

In conclusion, this advanced analysis validates the clinical potential of several models, particularly the fine-tuned Xception and InceptionResNetV2 architectures. It provides a quantitative basis for selecting not only the best model but also the optimal operating point (threshold) to balance the clinical requirements of sensitivity and specificity, thereby fulfilling a key objective of this investigation.

Future Work

Our unified-dataset strategy, colour-space enhancements (bilateral + CLAHE-Lab), and grayscale/colour ensemble pipelines achieved state-of-the-art performance while keeping computational cost within real-time constraints. Nevertheless, several avenues remain open to push both accuracy and deployability still further.

Architecture Exploration

Vision Transformers (ViT). Pure transformer backbones (and hybrid CNN-Transformer encoders) can model long-range morphological relationships between clustered cells, something convolution-only nets can miss. We will

benchmark vanilla ViT-B/16, Swin-Tiny, and ViT-PSO-SVM-style heads on our unified dataset and measure gains in recall for rare abnormal phenotypes.

Hover-Net vs U-Net for Segmentation. Hover-Net jointly predicts (i) nuclear pixel probability, (ii) horizontal & vertical distance maps, and (iii) instance masks, giving superior cell-instance separation in overlapped regions. Replacing our current U-Net preprocessing with Hover-Net could yield cleaner cytoplasm contours, reduce false positives in dense clusters, and feed higher-quality ROIs to the classifier.

Advanced Data Augmentation

GAN-Based Synthesis. We will train class-conditioned StyleGAN-v2 and Diffusion-based generators to create realistic minority-class samples and augment under-represented staining variations. This should not only balance classes but also regularise the decision boundary, increasing robustness to unseen labs.

Advanced Architectures — Residual Squeeze VGG16 (RS-VGG16)

The paper “Residual Squeeze VGG16 for Nail Disease Classification” introduces an architecture that aligns perfectly with our needs:

- **Residual (skip) connections**—borrowed from ResNet—allow gradients to flow unimpeded through deep stacks, mitigating vanishing-gradient issues.
- **Squeeze-and-Excitation (SE) blocks** act as channel-wise attention:
    - **Squeeze:** global average pooling collapses spatial information to a channel descriptor.
    - **Excitation:** a two-layer MLP learns the importance of each channel.

  The learned weights then recalibrate feature maps, amplifying informative channels and damping noisy ones.

Implementing RS-VGG16 will shrink the parameter count of our VGG16 backbone while likely boosting accuracy thanks to better channel attention. Time constraints prevented its inclusion in the present study.

Hierarchical and Expanded Datasets

Hierarchical CNNs (HCNNs). HCNNs mirror clinical workflow by aggregating context from multiple neighbouring cells, already surpassing 90 % accuracy in related cytology tasks. Integrating an HCNN branch on slide-level tiles could elevate whole-slide inference reliability.

Dataset Growth to Cx22 & BPT.

- Cx22 adds variation in staining protocols and imaging hardware.
- BPT offers 1920 × 1080 px high-resolution fields, ideal for pretraining segmentation encoders. Sophisticated data-engineering pipelines—patch sampling, stain-normalisation, and label harmonisation—will be built to fold these corpora into our unified set.

Multi task U-Net implementation

Due to mostly computational constraints, we weren’t able to train our model using the bottleneck for U-Net architecture. (We ran out of Google Colab credits)