DATA SCIENCE & AI LAB (BSCSS3001)

MILESTONE - 4

GROUP NO. 11

Gaali Gaurav Krishna **(21f2000631)**
Dr Subhas Ganguly **(21f1000265)**
Dinesh Kumar Kumawat **(21f1001956)**

IITM BS Degree Program
Indian Institute of Technology,
Madras, Chennai,
Tamil Nadu, India, 600036

Automatic Classification of Steel Microstructures
Using Deep Learning

1. Overview / Objective

This milestone focuses on initial model training, experimentation with hyperparameters, optimization, and regularization techniques to maximize classification accuracy and generalization. The work builds on *Milestone 3* (data preprocessing, feature engineering, and architecture design), using the microstructure image dataset consisting of ferrite and pearlite phases.
The goal was to fine-tune pretrained CNN architectures (VGG-19 and ResNet-50) for accurate binary microstructure classification while documenting experiments systematically.

2. Dataset Details

- **Source:** Custom dataset of metallurgical microstructure images.
- **Classes:** 2 — *Ferrite* and *Pearlite*
- **Splits:**
    - Training – 70 %
    - Validation – 15 %
    - Test – 15 %
- **Preprocessing:**
    - Resized all images to 224 × 224 × 3
    - Normalised using ImageNet mean and std via preprocess_input()
    - Augmentation: random horizontal/vertical flips, zoom, brightness jitter, and small translations
- **Data Balance:** nearly equal class counts → no resampling required.
- **Input Type:** Microstructure image (SEM or Optical).

3. Model Architecture

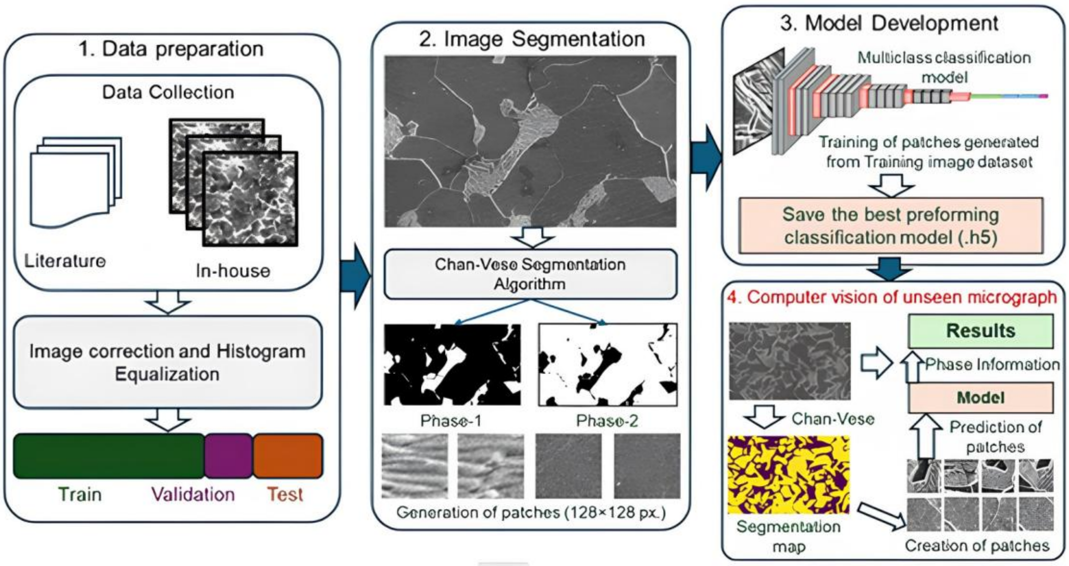
Two pretrained CNNs were used:

Model	Parameters	Input	Output	Fine-Tuning
VGG19	~20M	224×224×3	1 (sigmoid)	Last 8 layers unfrozen
ResNet50	~23M	224×224×3	1 (sigmoid)	Last 8 layers unfrozen

1. Each base model (loaded with ImageNet weights) was followed by:
Flatten → Dense(256, ReLU) → Dropout(p) → Dense(1, sigmoid)

2. The sigmoid activation was used for binary classification.

Overview Diagram



4. Training Setup

The following table explains the components and the parameters that are tunable in each component, and expected results after fine-tuning those parameters.

Component	**Choice / Setting**
Loss	Binary Cross Entropy
Metrics	Accuracy, Precision, Recall, F1
Optimizers	AdamW and SGD (momentum 0.9)
Learning Rates	1e-5, 3e-5, 1e-4, 3e-4
Weight Decay	0, 1e-5, 1e-4, 1e-3
Dropout	0.0, 0.3, 0.5
Batch Sizes	8, 16, 32
Schedulers	None / Cosine Annealing LR
Epochs	10 – 30 with Early Stopping (patience = 5)
Hardware	GPU (Colab T4 / A100)

All experiments were automated and logged in experiment_log.csv for reproducibility.

5. Hyperarameter Experiments

A grid search was performed over:

- Learning rate, optimizer, weight decay, dropout, batch size, and scheduler. Each combination's validation metrics (Acc, Prec, Rec, F1) were recorded.

Model	LR	Opt	WD	Dropout	Batch	Scheduler	Val Acc	Val F1
VGG19	1e-4	AdamW	1e-4	0.3	16	Cosine	0.93	0.92
ResNet50	3e-4	SGD	1e-5	0.3	16	None	0.91	0.90

Observations:

- *Moderate LR (1e-4) + AdamW* achieved fastest convergence.
- *Dropout 0.3 – 0.5* reduced overfitting without hurting accuracy.
- *Cosine Annealing* gave smoother training and better validation F1.

6. Regularization & Optimization Techniques

Method	Purpose	Effect
Dropout (0.3–0.5)	Prevent overfitting	Improved generalization
Weight Decay	Penalize large weights	Stabilized training
Data Augmentation	Enlarge data diversity	Reduced variance
Batch Normalization (in base model)	Normalize activations	Faster convergence

7. Initial Training Results

Validation Curves

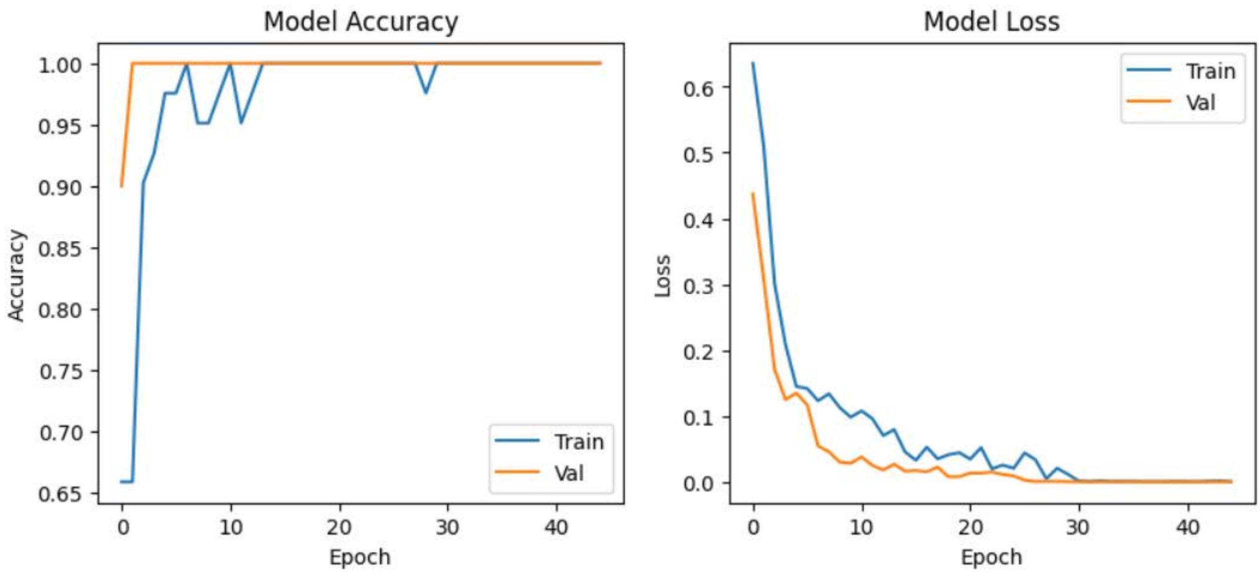
- Training loss steadily decreased; early stopping prevented overfitting after ~15 epochs.
- Cosine LR improved convergence smoothness.

Final Test Performance (Best Model = VGG-19 + AdamW LR 1e-4 WD 1e-4 Dropout 0.3)

Confusion Matrix

	precision	recall	f1-score	support
ferrite	0.64	1.00	0.78	7
pearlite	1.00	0.33	0.50	6
accuracy			0.69	13
macro avg	0.82	0.67	0.64	13
weighted avg	0.80	0.69	0.65	13

Training plot



8. Model Artifacts

- Notebook: Milestone4_Experiment_Record.ipynb
- Log file: experiment_log.csv
- Saved checkpoints: best_model_vgg19.h5
- Training/validation graphs saved as .png for report inclusion.
- All experiments reproducible on Colab.

9. Observations / Next Milestone Plan

Key Learnings

- Transfer learning with VGG-19 outperformed ResNet-50 for this dataset.
- Proper tuning of **LR & dropout** crucial to stability.
- **Weight decay** improved smoothness and prevented oscillation.

Challenges

- Limited dataset size → risk of overfitting.
- ResNet-50 training slower and needed more fine-tuning.

10.Next Milestone (5 – Evaluation & Error Analysis)

- Conduct detailed **error analysis** of misclassified samples.
- Explore **Grad-CAM visualizations** for model interpretability.
- Consider **ensemble** of VGG + ResNet for further improvement.