Pattern Recognition

Assignment 2
HAM10000 Dataset
Skin Lesion Classification and Segmentation

Hassan Tamer 7405
Sohail Waleed 7372
Hana Waleed 7599

Data Preprocessing and Duplicate Detection

The process begins with data loading, where image paths are collected from a directory and organized into a SkinDataset for efficient handling. Transformations such as resizing and normalization are applied to ensure compatibility with the pretrained MobileNetV2 model. During embedding extraction, MobileNetV2 is used to compute image embeddings by removing the classification head, processing them in batches with GPU acceleration and mixed precision for enhanced speed.

The resulting embeddings are saved as a .npy file for subsequent steps. In duplicate removal, embedding dimensions are reduced by averaging over spatial axes, normalized to facilitate cosine similarity calculations, and compared pairwise. Images with similarities above the defined threshold (0.93) are marked as duplicates, and only unique embeddings are retained, along with their corresponding image paths. The output includes text files listing removed and remaining image paths, along with a summary of the number of duplicates removed and images retained.

Key strengths of this approach include the utilization of GPU acceleration and mixed precision for faster processing, efficient batch-wise embedding extraction to prevent memory overflow, and a robust cosine similarity-based duplication detection mechanism.

This is done to remove data leakage as there may be duplicate images that are found in both test and validation splits.

The same approach was used to find exactly matching images (Fig. 1). Duplicate images in the dataset may cause data leakage between train and test sets.

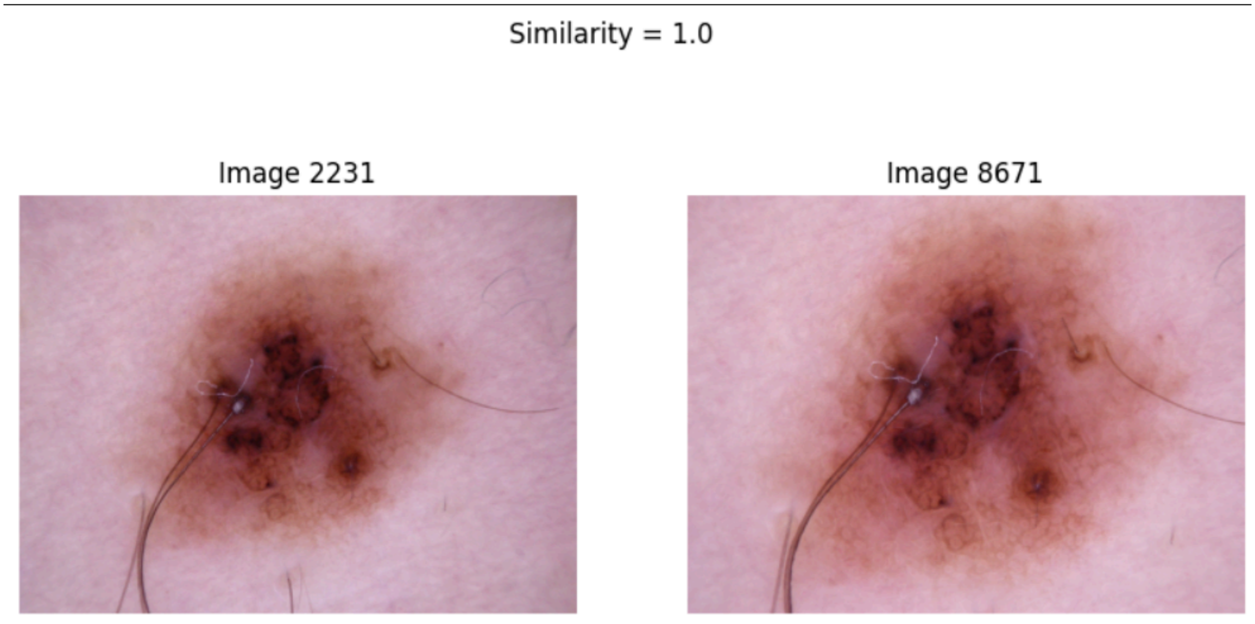


Fig.1 Exact Image found twice in the dataset

No data augmentation was done. As these skin lesion are very sensitive to color and maybe rotation in some cases and the dataset has already enough images.

Classification Model

Since the GroundTruth file contains labels for each image in one-hot encoded format, labels are converted into single-class identifiers by mapping the highest probability in the one-hot encoding to numeric indices (0–6). Then images are resized to 384×384, normalized, and converted into PyTorch tensors for model compatibility.
The following pie chart (Fig 2) illustrates the distribution of skin lesion classes.

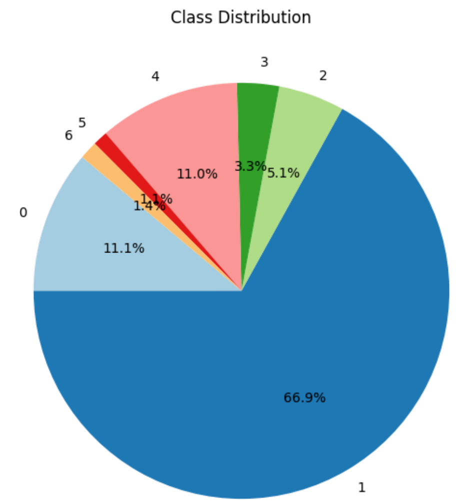


Fig 2. Class Balance in the Dataset

The dataset is divided into training, validation, and test sets in an 75/12.5/12.5 ratio. And as observed from the above pie chart , the classes are extremely unbalanced, which is going to affect the performance of the model. We tried oversampling but that usually caused the program to go out of memory and crash.

Architecture

We used the resnet50 pre-trained model for the classification. Due to its reliability and ability of rich feature extraction. It is well documented and proved its efficiency in medical tasks.

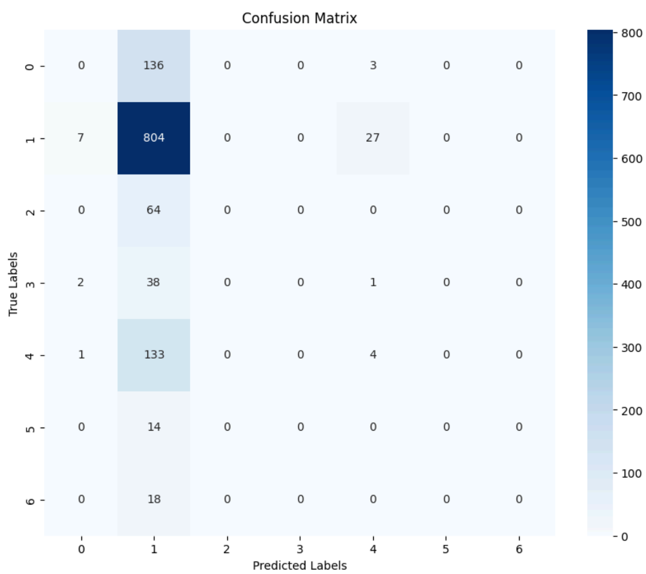
Using transfer learning , the weights could reach convergence faster.

```
Total parameters: 11,180,103
Trainable parameters: 11,180,103
```

Training

CrossEntropyLoss is used for classification.The Adam optimizer is configured with a learning rate of 5e−5.
The model is trained for 3 epochs with a batch size of 32.

Classification Metrics



Test Loss: 50.7477, Accuracy: 64.54%

Fig 3. Confusion Matrix for Test Set

Accuracy, precision, recall, and F1-score are calculated on validation and test sets as shown in (Fig 4). The model achieves approximately 66.93% accuracy on validation and 66.69% on test data. However, low precision and F1-score indicate potential issues, such as label imbalance or poor feature representation.

Classification Report:

	precision	recall	f1-score	support
Class 0	0.00	0.00	0.00	139
Class 1	0.67	0.96	0.79	838
Class 2	0.00	0.00	0.00	64
Class 3	0.00	0.00	0.00	41
Class 4	0.11	0.03	0.05	138
Class 5	0.00	0.00	0.00	14
Class 6	0.00	0.00	0.00	18
accuracy			0.65	1252
macro avg	0.11	0.14	0.12	1252
weighted avg	0.46	0.65	0.53	1252

Fig 4. Classification Metrics on the Test Set

Segmentation Model

This model was mainly for the segmentation task. We managed to attach a classification head so the model is able to do both segmentation and classification tasks using the

Model Evaluation

For the evaluation, the performance of segmentation and classification models is assessed using the following metrics:

- Segmentation Evaluation: Computes Dice scores and IOU for segmentation and visualizes predictions alongside ground truth.
- Classification Evaluation: Provides metrics like accuracy, precision, recall, and F1-score, complemented by a confusion matrix heatmap.
- Combined Evaluation: Integrates segmentation and classification tasks, ensuring alignment in outputs.

Architecture

We used the deeplabv3 model for the segmentation. It has a limited number of parameters, 42M parameters, while performing very well in segmentation tasks by assigning each pixel to a class. It is suitable for medical imaging tasks , pretrained, easy to fine tune and provides high accuracy.

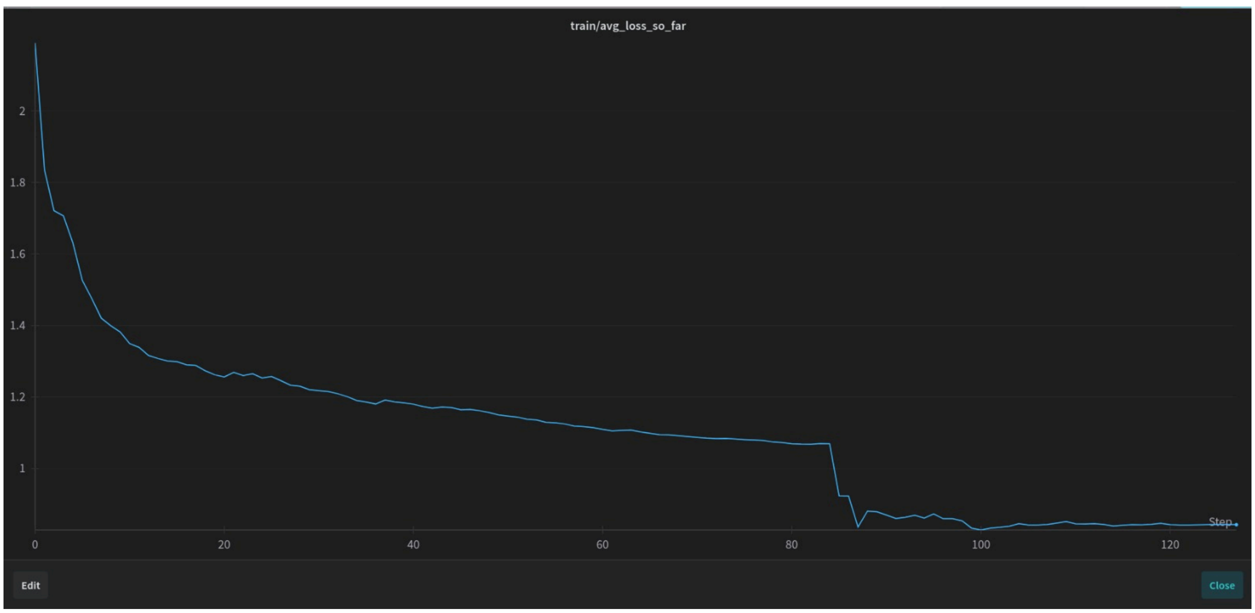
```
Total number of parameters in the model: 42,004,074
Using device: cuda
```

Training

Cross entropy Loss is used. The Adam optimizer is configured with a learning rate of 1×10^−4. The model is trained for 10 epochs with a batch size of 4.

Segmentation Metrics

Average IOU: 0.8519
Average Dice Score: 0.9182



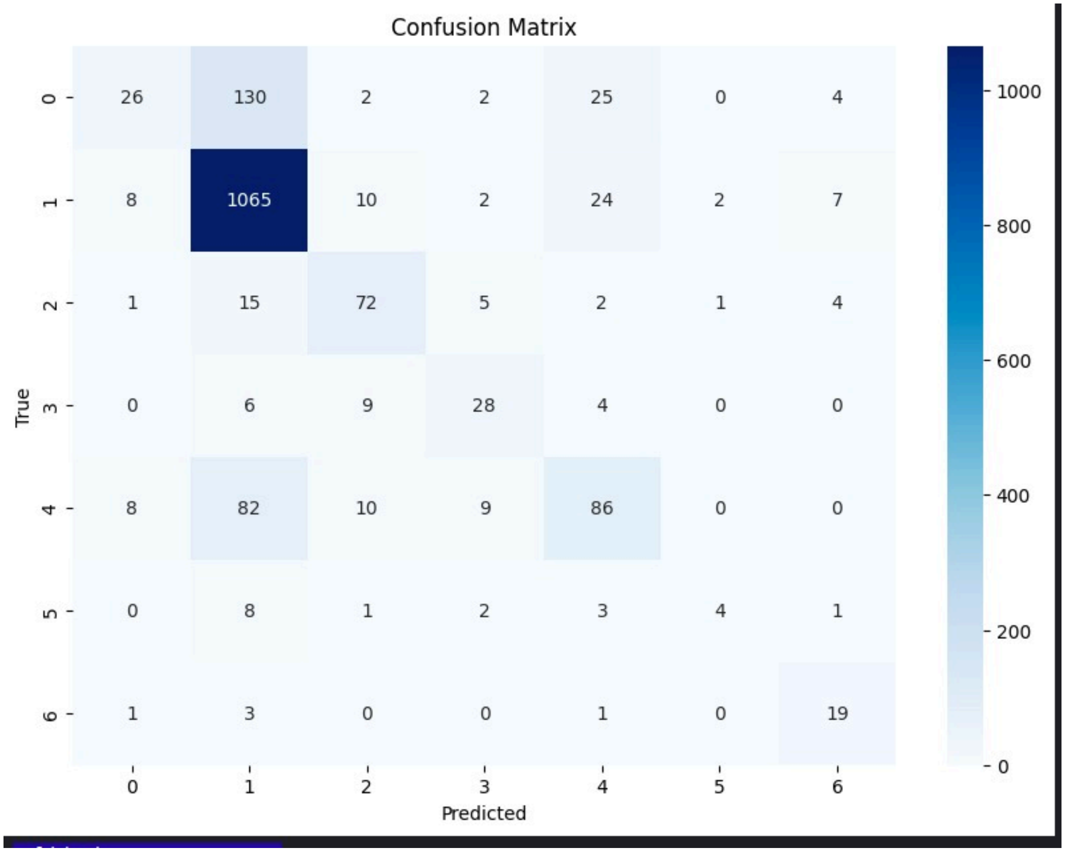
Validation Loss: 0.6461

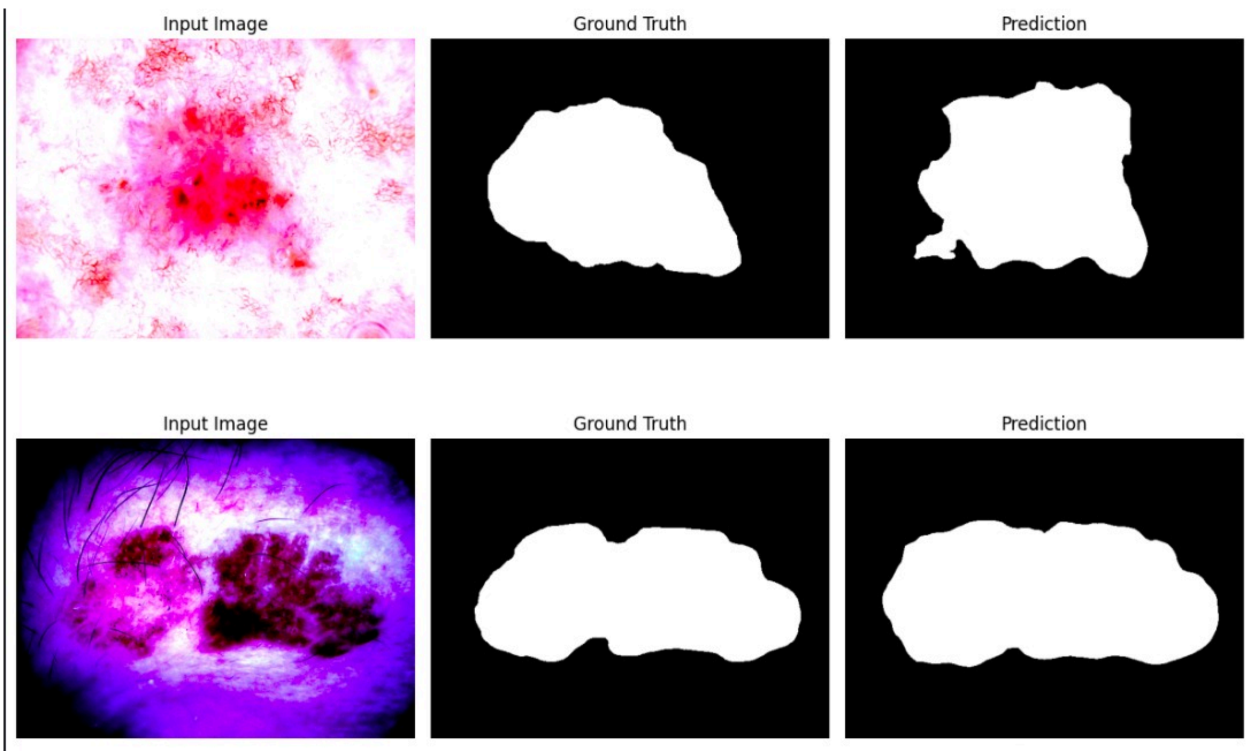
Accuracy: 76.83%

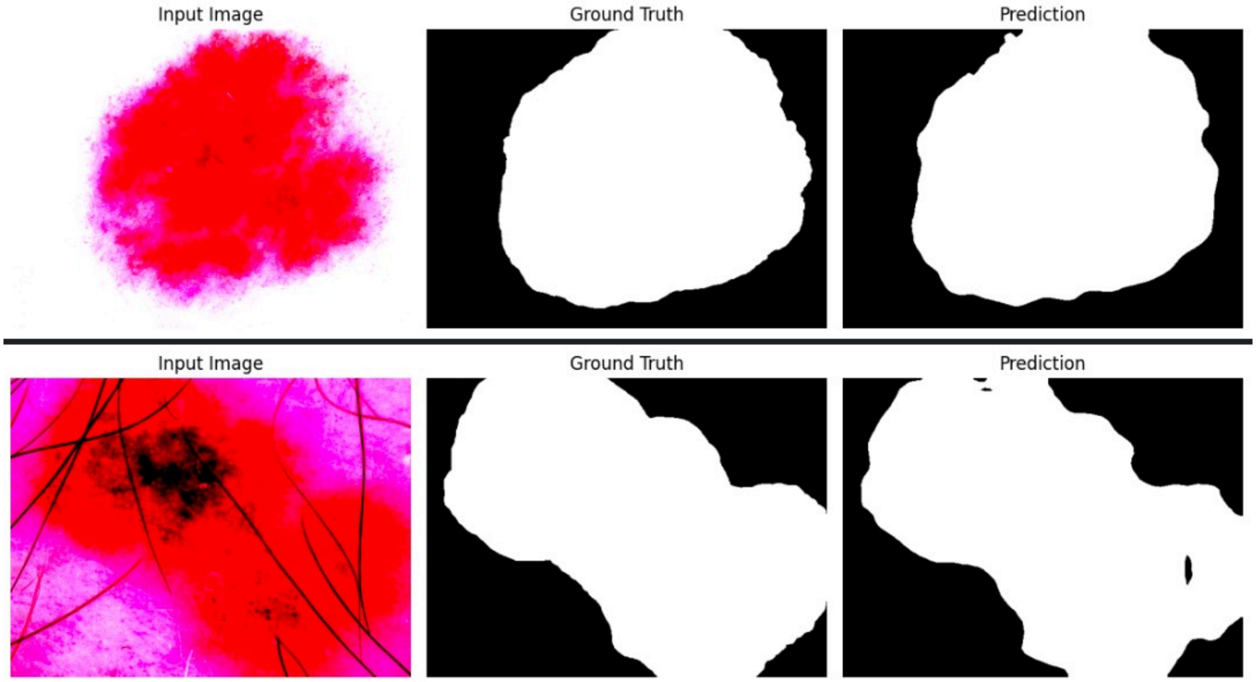
Precision: 0.7432

Recall: 0.7683

F1-Score: 0.7338


Confusion Matrix
True
Predicted
0
1
2
3
4
5
6
1000
800
600
400
200
0


Input Image
Ground Truth
Prediction
Input Image
Ground Truth
Prediction


Input Image
Ground Truth
Prediction
Input Image
Ground Truth
Prediction