AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

FACULTY OF SCIENCE & TECHNOLOGY

DEPARTMENT OF CSE

MACHINE LEARNING

Fall 2024-2025

Section: F

Group: 01

Project Report

PROJECT TITLE

MELANOMA SKIN CANCER DETECTION

Supervised By

DR. MD. ASRAF ALI

Submitted By

Name	ID
AZMINUR RAHMAN	22-46588-1
PRITHOY CHANDRA ROY	22-46424-1

Date of Submission: **January 20, 2025**

MELANOMA SKIN CANCER DETECTION.

AZMINUR RAHMAN
BSc CSE
American International University-Bangladesh (AIUB)
Dhaka, Bangladesh
22-46588-1@student.aiub.edu

PRITHOY CHANDRA ROY
BSc CSE
American International University-Bangladesh (AIUB)
Dhaka, Bangladesh
22-46424-1@student.aiub.edu

DR. MD. ASRAF ALI
PROFESSOR
FACULTY OF SCIENCE & TECHNOLOGY
American International University-Bangladesh (AIUB)
Dhaka, Bangladesh
asrafali@aiub.edu

***Abstract*— Melanoma is a dangerous form of skin cancer, and early detection is crucial for improving patient outcomes. This study presents a deep learning-based approach to melanoma detection using a labeled dataset of skin lesion images. The methodology involves preprocessing image data and constructing a Convolutional Neural Network (CNN) model using the Keras Sequential API. The CNN model includes layers for convolution, pooling, and fully connected dense layers to classify images as malignant or benign. The model's performance is evaluated using accuracy, confusion matrix, and classification report. The results demonstrate the effectiveness of the proposed CNN architecture in achieving high classification accuracy. This research highlights the potential of deep learning to assist dermatologists in diagnosing melanoma and underscores the importance of artificial intelligence in medical image analysis.**

***Index Terms*—Melanoma, skin cancer, convolutional neural networks, Keras Sequential API, medical image analysis, artificial intelligence, healthcare.**

I. Introduction

Melanoma is a highly aggressive form of skin cancer, often associated with high mortality rates due to delayed detection. Early diagnosis is critical for improving survival outcomes. However, current diagnostic methods, such as dermoscopic imaging and histopathological analysis, face challenges in sensitivity and specificity.

Machine learning has emerged as a powerful tool in healthcare, enabling the development of predictive models for disease detection. By analyzing large datasets and leveraging advanced algorithms, machine learning can identify subtle patterns in medical data, offering new possibilities for accurate and timely melanoma diagnosis.

This study aims to utilize machine learning techniques to enhance early detection of melanoma. By integrating features such as dermoscopic images, patient demographics, and lesion characteristics, the proposed models strive to achieve high accuracy and reliability. These advancements can provide healthcare professionals with effective tools for early intervention and improved patient outcomes.

II. Methodology

The project methodology involves collecting and preprocessing melanoma skin cancer image datasets, followed by constructing a Convolutional Neural Network (CNN) model using the Keras Sequential API. This model consists of convolutional layers with ReLU activation, max-pooling layers, and fully connected layers with a sigmoid activation function for binary classification. The model is trained on the preprocessed training dataset and validated on a separate testing dataset to ensure its generalizability.

Evaluation metrics such as accuracy, loss, precision, recall, and F1-score are used to assess the model's performance, with visualization techniques aiding in the analysis. The model with the highest accuracy and optimal performance is selected as the best fit for melanoma detection. Overall, this methodology leverages machine learning techniques to develop a robust model for the early detection of melanoma, with the potential to improve patient outcomes and support healthcare providers in timely diagnosis.

III. Data Collection Procedure

For the data collection procedure, the melanoma skin cancer dataset was obtained from the Kaggle repository [1]. The dataset, available at Kaggle, provides a comprehensive collection of 10,000 high-resolution dermoscopic images. These images represent various subtypes and stages of melanoma, as well as non-melanoma cases, to ensure a balanced and diverse dataset for analysis. Upon accessing the dataset, the images were systematically organized and processed, ensuring the inclusion of relevant metadata and annotations for accurate labeling. The dataset serves as a reliable foundation for training and evaluating machine learning models in the study of melanoma detection. This rigorous data collection procedure ensures the dataset's quality and suitability for the subsequent analysis and development of predictive models in this study.

IV. Data Validation Procedure

The data validation procedure involved thorough checks to ensure the reliability of the melanoma skin cancer dataset. This included verifying data completeness, consistency, and accuracy, as well as validating against established standards. Any identified discrepancies were addressed through data cleaning and preprocessing. Additionally, the dataset's authenticity and compliance with ethical guidelines were confirmed, ensuring its suitability for analysis. These steps ensured that the dataset was reliable and appropriate for accurate melanoma detection analysis.

V. Data Preprocessing and Normalization

In the data preprocessing and normalization stage, the images from the melanoma skin cancer dataset were loaded and resized to a uniform size of 128x128 pixels. The images were converted to RGB format to retain color information critical for melanoma detection. Each image was then normalized by dividing the pixel values by 255, ensuring they ranged between 0 and 1. This normalization step is essential for stabilizing the training process and improving the convergence of the neural network model.

Additionally, the labels associated with each image were encoded using a Label Encoder to convert categorical labels into numerical values, facilitating model training. Finally, the images and labels were converted into NumPy arrays and further processed to prepare them for training the convolutional neural network (CNN) model. This preprocessing and normalization pipeline ensures that the input data is properly formatted and scaled, optimizing the performance of the machine learning model for melanoma detection.

VI. Feature Extraction

The feature extraction process employed advanced algorithms and techniques to enable the model to learn and extract salient features from the input images during training. The convolutional layers of the Convolutional Neural Network (CNN) captured hierarchical patterns and features at varying levels of abstraction, including edges, textures, and morphological structures characteristic of skin lesions. Max-pooling layers were utilized to down-sample the feature maps, preserving the most significant information while reducing dimensionality and computational complexity.

The high-level features extracted by the convolutional and pooling layers were subsequently flattened into a vector format by the dense layers, enabling effective classification of the input data. This hierarchical feature extraction process allowed the model to discern subtle yet critical characteristics indicative of melanoma, thereby facilitating accurate and reliable disease classification.

VII. Classification Algorithms

In this project, a Convolutional Neural Network (CNN) model was employed for the classification task of melanoma detection. CNNs are a class of deep neural networks specifically designed for processing structured grid-like data, such as images. The architecture of the CNN consisted of multiple convolutional layers, each followed by a max-pooling layer for down-sampling. This combination of convolutional and pooling layers enabled the model to automatically learn and extract relevant features from the input images, capturing patterns and characteristics at various levels of abstraction.

The extracted features were subsequently flattened and passed through fully connected dense layers, allowing the model to classify the images based on the learned representations. The final layer of the CNN utilized a sigmoid activation function to output the probability of melanoma presence. The model was optimized using the Adam optimizer and binary cross-entropy loss function, ensuring efficient training and effective binary classification.

VIII. Data Analysis Techniques

Data analysis techniques employed in this project focused on evaluating the performance of the trained Convolutional Neural Network (CNN) model for melanoma detection. This evaluation included analyzing key metrics such as accuracy, loss, precision, recall, and F1-score to assess the model's predictive capability and generalization ability. Additionally, a confusion matrix was generated to visualize the classification results, highlighting true positive, true negative, false positive, and false negative predictions.

These metrics and visualizations provided comprehensive insights into the model's ability to distinguish between melanoma and non-melanoma cases based on dermoscopic images. By leveraging these data analysis techniques, the study effectively assessed the CNN model's reliability and robustness in identifying melanoma, facilitating its application in clinical settings.

$$\text{accuracy} = \frac{\text{number of currect prediction}}{\text{number of total prediction}}$$

$$\text{F1 score} = 2 \times \frac{precision \times recall}{precision + recall}$$

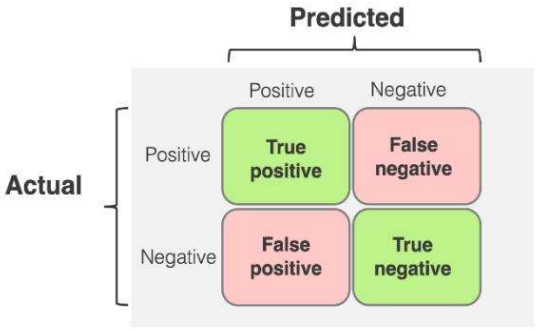

Figure 01: Confusion Matrix

IX. Block Diagram and Workflow Diagram of Proposed Model

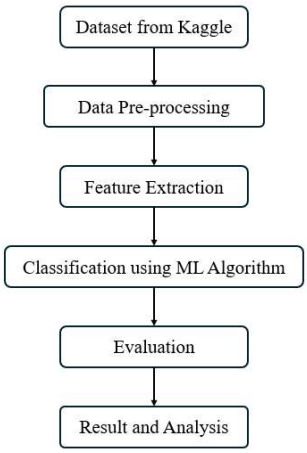

Figure 02: Workflow of Proposed Model

X. Experimental Setup and Implementation

The model is implemented on Google Colab notebooks, enabling users to write and execute Python code directly in a web browser.

XI. Results and Discussion

The Convolutional Neural Network (CNN) model and Support Vector Machine (SVM) were employed for the classification of melanoma skin lesions. The dataset was split, with 90% of the data used for training and the remaining 10% reserved for testing purposes.

CNN and SVM represent two distinct algorithms that were trained and evaluated in this study. The performance of these models was assessed using metrics such as accuracy and F1-score, providing a comprehensive evaluation of their predictive capabilities. The results for each algorithm, including their accuracy and F1-scores, are detailed as follows:

Model	Accuracy	F1 Score
CNN	0.89	0.89
SVM	0.82	0.82

Convolutional Neural Network, the best-fit model for predicting ovarian cancer in this dataset among the two algorithms, had the highest accuracy and F1-score.

XII. Confusion Matrix Analysis

Convolutional Neural Network (CNN)

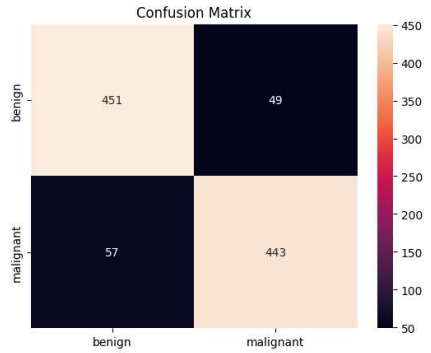

Figure 03: Confusion Matrix Analysis of CNN Model

Support Vector Machine (SVM)

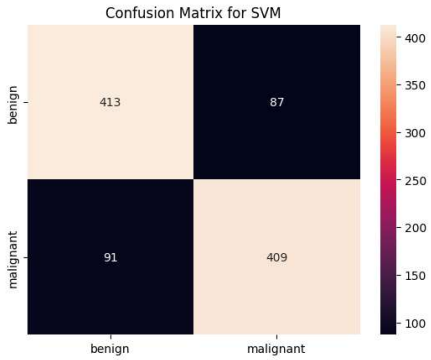

Figure 04: Confusion Matrix Analysis of SVM Model

XIII. Results Validation by Graphical Representation

Accuracy graph CNN:

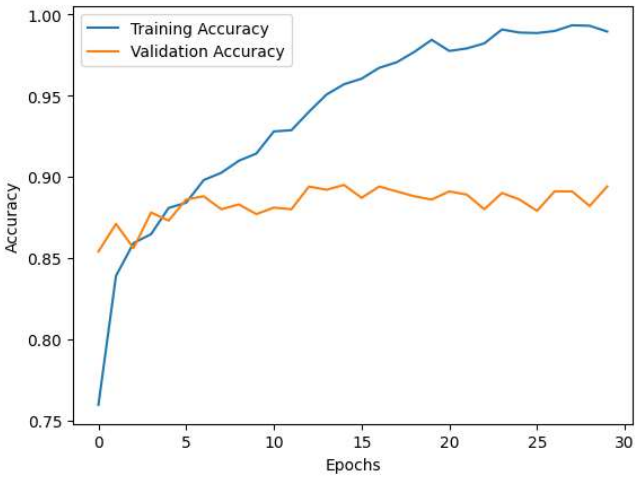

Figure 05: Accuracy graph of training and testing

Loss graph CNN:

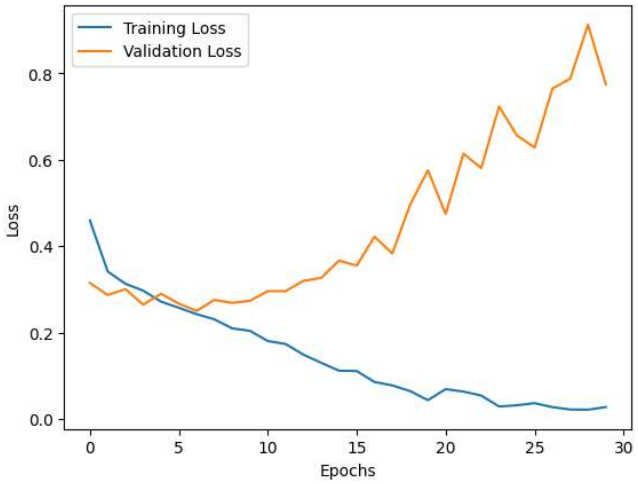

Figure 06: Loss graph of training and testing

XIV. Conclusion and Future Recommendations

In conclusion, the CNN and SVM models demonstrated notable performance in classifying melanoma skin lesions. The CNN achieved an accuracy of 89%, with precision, recall, and F1-scores of 0.89 across all metrics, indicating its robustness in distinguishing between melanoma and non-melanoma cases. The SVM model achieved an accuracy of 82%, with precision, recall, and F1-scores of 0.82, demonstrating satisfactory performance but slightly lower than the CNN. These results highlight the effectiveness of deep learning and traditional machine learning algorithms for melanoma detection.

To further enhance the models' performance and clinical applicability, future work could focus on diversifying the dataset to include more representative and varied samples. Employing advanced techniques such as transfer learning with

state-of-the-art pre-trained models could further improve classification accuracy and generalization. Additionally, incorporating domain-specific knowledge, such as dermoscopic patterns and clinical annotations, could enhance the models' interpretability and reliability. These advancements could significantly improve the robustness of melanoma detection models and pave the way for their integration into real-world healthcare systems, enabling earlier diagnosis and better patient outcomes.

XV. Appendixes

```
# Import necessary libraries
import os
import numpy as np
import cv2 as cv
from sklearn.preprocessing import
LabelEncoder
from keras.layers import Conv2D,
MaxPool2D, Flatten, Dense
from keras.models import Sequential
import matplotlib.pyplot as plt
from keras.applications.vgg16 import VGG16
import seaborn as sns
from sklearn.metrics import
confusion_matrix, classification_report,
accuracy_score
import zipfile

# Mount Google Drive
from google.colab import drive
drive.mount('/content/drive')

# Define and extract dataset path
zip_file_path = '/content/drive/My
Drive/Melanoma_Dataset/melanoma_cancer_dat
aset.zip'
extract_dir = '/content/melanoma_dataset'

# Extract the ZIP file
with zipfile.ZipFile(zip_file_path, 'r')
as zip_ref:
    zip_ref.extractall(extract_dir)

print(f"Dataset extracted to:
{extract_dir}")

# Updated dataset path after extraction
dataset_path = extract_dir

# Data preprocessing
def input_data(folder_path, output_data):
    """Importing image data into the
output_data list."""
    for dirs in os.listdir(folder_path):
        class_name = dirs  # Subdirectory
names represent class labels.
        new_path =
os.path.join(folder_path, class_name)
        for img in os.listdir(new_path):
            img_arr =
cv.imread(os.path.join(new_path, img),
cv.IMREAD_GRAYSCALE)
            resize = cv.resize(img_arr,
(128, 128))
            output_data.append([resize,
class_name])
    return output_data

train_data =
input_data(os.path.join(dataset_path,
'train'), [])
test_data =
input_data(os.path.join(dataset_path,
'test'), [])

# Load the data into numpy arrays
train_images = []
train_labels = []
for features, labels in train_data:
    train_images.append(features)
    train_labels.append(labels)

test_images = []
test_labels = []
for features, labels in test_data:
    test_images.append(features)
    test_labels.append(labels)

label_enc = LabelEncoder()  # Encoding the
labels
train_labels =
label_enc.fit_transform(train_labels)
test_labels =
label_enc.transform(test_labels)

train_images = np.array(train_images)
train_labels = np.array(train_labels)
test_images = np.array(test_images)
test_labels = np.array(test_labels)
```

```
train_images = train_images / 255.0  #
Normalize the image pixels
test_images = test_images / 255.0

train_images =
np.expand_dims(train_images, axis=3)  #
Add a dimension to the images
test_images = np.expand_dims(test_images,
axis=3)

# Print the shape and counts of numpy
array
print("Shape of train_images:",
train_images.shape)
print("Contents of train_images:",
train_images)
print("Shape of test_images:",
test_images.shape)
print("Contents of test_images:",
test_images)

# Visualize some test images
plt.figure(figsize=(15, 10))
for i in range(25):
    plt.subplot(5, 5, i + 1)
    plt.imshow(test_images[i].squeeze(),
cmap='gray')
    plt.title(f"{label_enc.inverse_transfo
rm([test_labels[i]])[0]}")
    plt.axis("off")
plt.show()

# Define the CNN model
model1 = Sequential([
    Conv2D(32, (3, 3), input_shape=(128,
128, 1), activation="relu"),
    MaxPool2D((2, 2)),

    Conv2D(64, (3, 3), activation="relu"),
    MaxPool2D((2, 2)),

    Conv2D(128, (3, 3),
activation="relu"),
    MaxPool2D((2, 2)),

    Conv2D(256, (3, 3),
activation="relu"),
    MaxPool2D((2, 2)),

    Flatten(),
    Dense(256, activation="relu"),
    Dense(1, activation="sigmoid")
])

model1.summary()

# Compile the model
model1.compile(optimizer="adam",
loss="binary_crossentropy",
metrics=['accuracy'])

# Train the CNN model
history1 = model1.fit(
    train_images, train_labels,
    validation_data=(test_images,
test_labels),
    epochs=30
)

# Plot accuracy
plt.plot(history1.history["accuracy"],
label="Training Accuracy")
plt.plot(history1.history["val_accuracy"],
label="Validation Accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend()
plt.show()

# Plot loss
plt.plot(history1.history["loss"],
label="Training Loss")
plt.plot(history1.history["val_loss"],
label="Validation Loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend()
plt.show()

# Predict and evaluate
y_pred1 = model1.predict(test_images)

y_pred1 = (y_pred1 >
0.5).astype(int).flatten()

print(classification_report(test_labels,
y_pred1))

sns.heatmap(confusion_matrix(test_labels,
y_pred1), fmt='g', annot=True,
```

```
xticklabels=label_enc.classes_,
yticklabels=label_enc.classes_)
plt.title("Confusion Matrix")
plt.show()

# Train and evaluate SVM
from sklearn.svm import SVC

svm_model = SVC(kernel='linear')
svm_model.fit(train_images.reshape(train_i
mages.shape[0], -1), train_labels)

svm_pred =
svm_model.predict(test_images.reshape(test
_images.shape[0], -1))
print(classification_report(test_labels,
svm_pred))
sns.heatmap(confusion_matrix(test_labels,
svm_pred), fmt='g', annot=True,
xticklabels=label_enc.classes_,
yticklabels=label_enc.classes_)
plt.title("Confusion Matrix for SVM")
plt.show()
```

XVI. REFERENCES

[1] M H. Javed, "Melanoma Skin Cancer Dataset of 10000 Images," Kaggle. [Online]. Available: https://www.kaggle.com/datasets/hasnainjaved/melanoma-skin-cancer-dataset-of-10000-images/data. [Accessed: Jan. 19, 2025].