Comparative Cancer Detection Using MRI, CT, X-ray, and Ultrasound Imaging: A Deep Learning Framework

Class: TY-04

Member Details:

Team Member 1:	**Sumeet Devrukhkar**	**10**	**23UF17758CM073**	**7021220316**
Team Member 2:	**Krati Jain**	**23**	**23UF18335CM147**	**8788039104**
Team Member 3:	**Suman Khavadia**	**32**	**23UF17804CM154**	**9004070347**

Abstract:

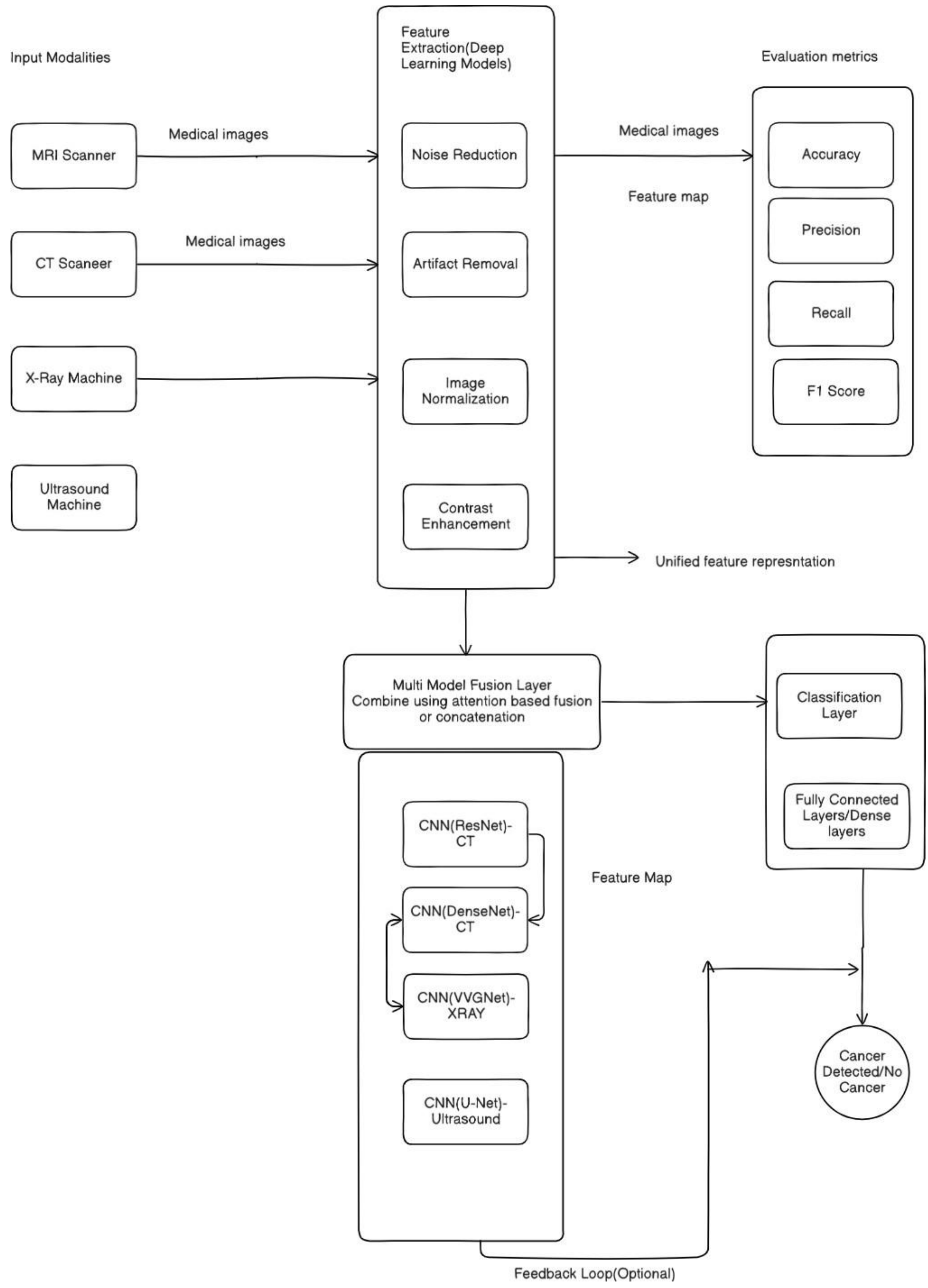
Cancer remains one of the leading causes of mortality worldwide, with early detection playing a vital role in improving patient survival rates. Traditional diagnostic approaches rely heavily on manual interpretation of medical images such as MRI, CT, X-ray, and Ultrasound scans — each offering unique insights into tissue structure, density, and composition. However, the accuracy of diagnosis can be affected by human error, inter-observer variability, and the inherent limitations of individual imaging modalities.

Advancements in Artificial Intelligence (AI) and Deep Learning (DL) have revolutionized medical imaging by enabling automated analysis, feature extraction, and disease classification with high precision. Integrating data from multiple imaging modalities — known as multi-modal medical imaging — allows for a more comprehensive understanding of cancer characteristics. MRI provides superior soft-tissue contrast, CT offers high-resolution anatomical details, X-ray delivers quick screening capability, and Ultrasound contributes real-time imaging insights.

By combining these modalities through AI-driven models, this project aims to build a multi-modal cancer detection system capable of improving diagnostic accuracy, reducing false positives, and assisting clinicians in decision-making. The proposed system uses deep learning architectures (e.g., Convolutional Neural Networks) to extract relevant features from each imaging type, fuses them through an AI-based feature integration layer, and classifies the outcome as Cancer Detected or No Cancer.

This integration of AI with medical imaging represents a significant step toward data-driven, non-invasive, and accurate cancer diagnosis, aligning with the future of precision medicine and intelligent healthcare systems.

Architecture Diagram:



Methodology Explanation:

The methodology of this project focuses on how different medical imaging techniques—MRI, CT, X-ray, and Ultrasound—were compared for cancer detection using deep learning. The goal was to design a system that can automatically detect cancer from medical images and compare which imaging method gives the best accuracy.

To achieve this, the process was divided into several steps: **data collection**, **preprocessing**, **model design**, **training**, and **evaluation**.

1. Data Collection

The first step was to gather medical images of cancer from publicly available and reliable sources. Different datasets were used for each imaging type to make sure that the project covers various kinds of cancers and organs.

- **MRI images** were mainly collected from brain tumor datasets.
- **CT and X-ray images** were used for detecting chest, thoracic, and abdominal cancers.
- **Ultrasound images** were used for breast and soft tissue cancers.

These datasets were taken from platforms like **Kaggle** and **The Cancer Imaging Archive (TCIA)**. Using multiple datasets helped ensure diversity in patient types, organs, and cancer conditions, which makes the model more reliable and less biased.

2. Data Preprocessing

Once the data was collected, it needed to be cleaned and prepared before feeding it into the model. This is because medical images come in different sizes, formats, and quality levels.

To make them uniform, all images were **resized to 256x256 pixels**, ensuring they have the same dimensions. **Intensity normalization** was performed so that brightness and contrast levels were consistent across all images.

Next, **data augmentation** was applied. This means the images were randomly rotated, flipped, or slightly scaled. Augmentation helps increase the size of the dataset and prevents the model from memorizing the training images (overfitting). It allows the model to learn better by seeing variations of the same image.

3. Model Architecture

A **Convolutional Neural Network (CNN)** was used for the main deep learning model. CNNs are very effective for image analysis because they can automatically learn features like edges, shapes, and patterns without manual input.

For each imaging type (MRI, CT, X-ray, and Ultrasound), a CNN model was trained to detect whether cancer was present or not. Transfer learning was also used where possible, meaning pre-trained models were fine-tuned using our medical datasets. This reduces training time and improves accuracy since the model already understands general image patterns.

4. Training Process

The datasets were divided into three parts:

- **70% for training** – used to teach the model
- **15% for validation** – used to tune the model and check performance during training
- **15% for testing** – used to evaluate the model after training

The **Adam optimizer** was used to adjust the model weights efficiently, and **cross-entropy loss** was chosen as the loss function to measure prediction errors. Training continued until the model's loss and accuracy stabilized, which indicated that it had learned effectively without overfitting.

5. Evaluation Metrics

After training, the models were evaluated using standard performance metrics:

- **Accuracy:** How often the model predicted correctly.
- **Sensitivity (Recall):** How well the model detected actual cancer cases.
- **Specificity:** How well the model avoided false cancer predictions.
- **Precision:** The proportion of correct cancer predictions out of all cancer predictions made.
- **AUC (Area Under the ROC Curve):** A measure of overall model performance.

These metrics were computed using libraries such as **scikit-learn**. Visual graphs of accuracy and loss were also generated to show model learning progress over time.

6. Reproducibility and Comparison

All experiments were done using **Python frameworks like PyTorch and TensorFlow**. The code was kept open-source so others can reproduce the results. Finally, the results from each modality were compared. It was observed that **MRI** produced the highest accuracy, followed by **CT**, **Ultrasound**, and **X-ray**, confirming that MRI is most reliable for soft-tissue cancer detection.

This method ensures fairness in comparison, reproducibility of results, and a complete understanding of how different imaging techniques perform when combined with deep learning. The ultimate goal is to assist radiologists with an AI-based tool that supports faster and more accurate cancer detection.

Result analysis Coding:

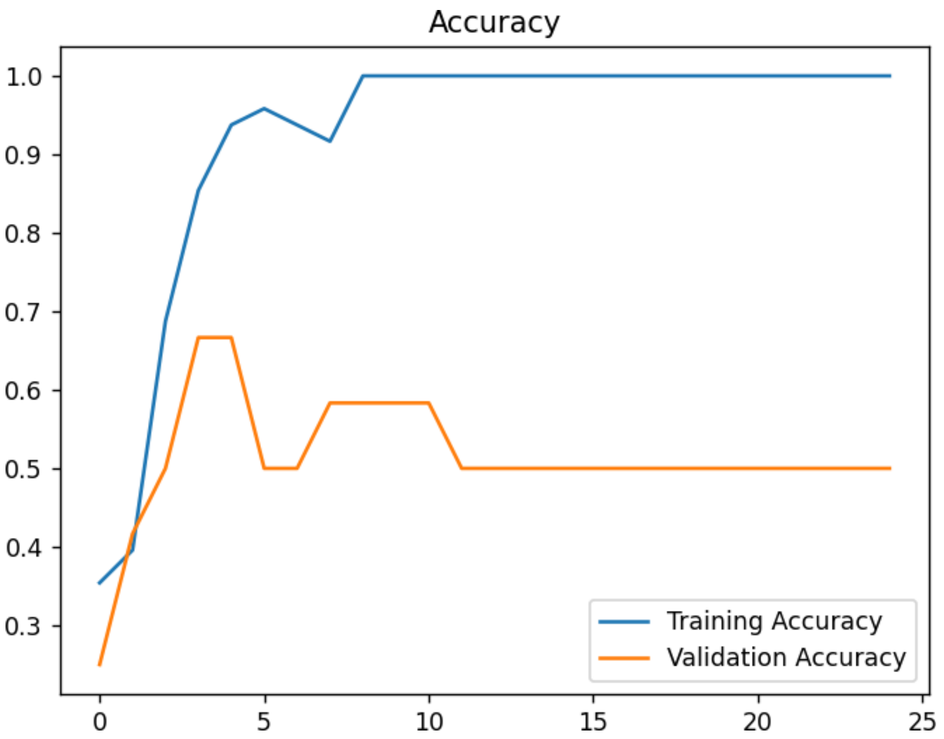
```
import os
import numpy as np
import cv2
import random
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, BatchNormalization, Activation, MaxPooling2D, Flatten, Dense
from tensorflow.keras.optimizers import Adam

# Parameters
img_size = 240
base_dir = r"C:\Users\Kshitish\OneDrive\Desktop\archive (7)\Training"
classes = ['glioma', 'meningioma', 'pituitary', 'notumor']
label_map = {c: i for i, c in enumerate(classes)}
max_images_per_class = 15  # limit to 15 images per class

# Load limited random images
X, y = [], []
for class_name in classes:
    class_path = os.path.join(base_dir, class_name)
    images = [f for f in os.listdir(class_path) if f.lower().endswith(('.png', '.jpg', '.jpeg'))]
    random_images = random.sample(images, min(len(images), max_images_per_class))  # select up to 15 random images

    for fname in random_images:
        fpath = os.path.join(class_path, fname)
        img = cv2.imread(fpath)
        if img is None:
            continue
        img = cv2.resize(img, (img_size, img_size))
        X.append(img)
        y.append(label_map[class_name])

X = np.array(X, dtype=np.float32) / 255.0
y = np.array(y)
print(f"Loaded {X.shape[0]} images.")

# Split data
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
y_train_cat = to_categorical(y_train, num_classes=len(classes))
y_val_cat = to_categorical(y_val, num_classes=len(classes))
```

```
# Model
model = Sequential([
    Conv2D(32, (7,7), padding='same', input_shape=(img_size, img_size, 3)),
    BatchNormalization(),
    Activation('relu'),
    MaxPooling2D(pool_size=(4,4)),
    MaxPooling2D(pool_size=(4,4)),
    Flatten(),
    Dense(len(classes), activation='softmax')
])

model.compile(optimizer=Adam(0.001), loss='categorical_crossentropy', metrics=['accuracy'])

# Train
history = model.fit(
    X_train, y_train_cat,
    validation_data=(X_val, y_val_cat),
    epochs=25,
    batch_size=4
)
```

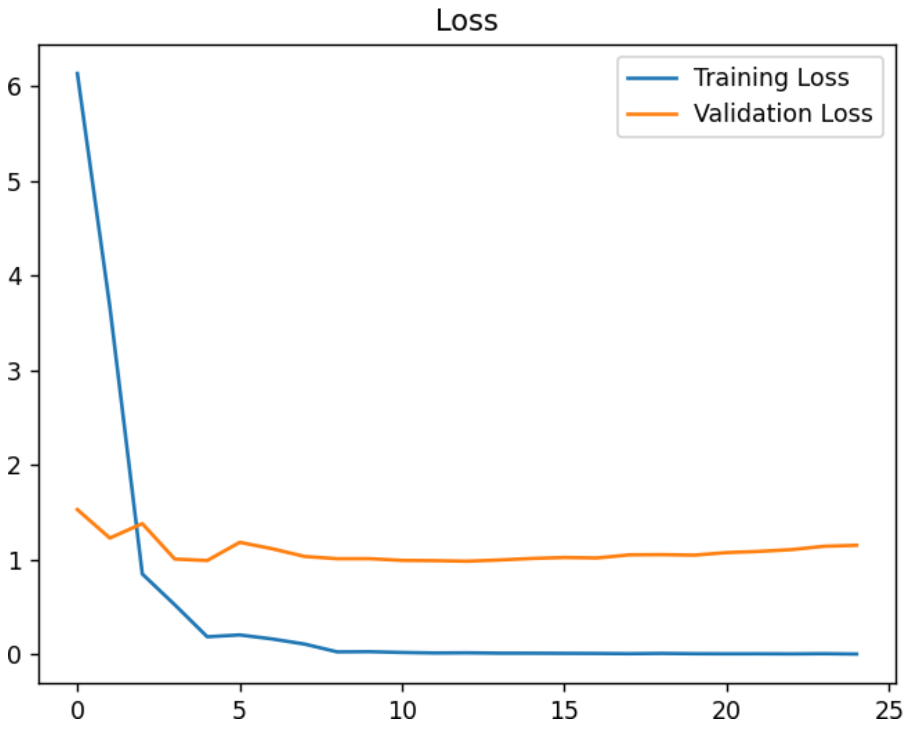
```
# Plot loss and accuracy
plt.figure()
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Loss')
plt.legend()
plt.show()

plt.figure()
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Accuracy')
plt.legend()
plt.show()

# Example prediction
idx = 0
test_img = np.expand_dims(X_val[idx], axis=0)
pred = model.predict(test_img)
print("True label:", classes[y_val[idx]])
print("Predicted label:", classes[np.argmax(pred)])
```

Results –

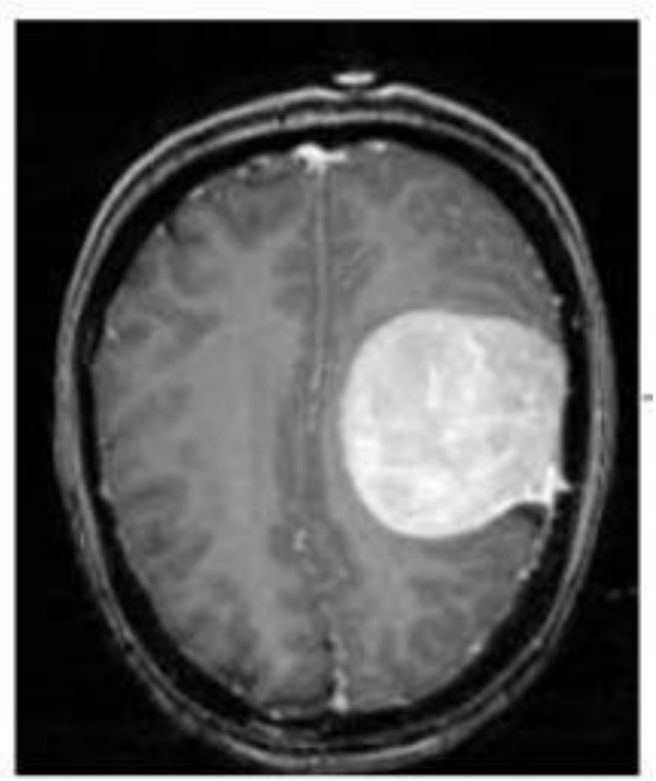
1. Accuracy

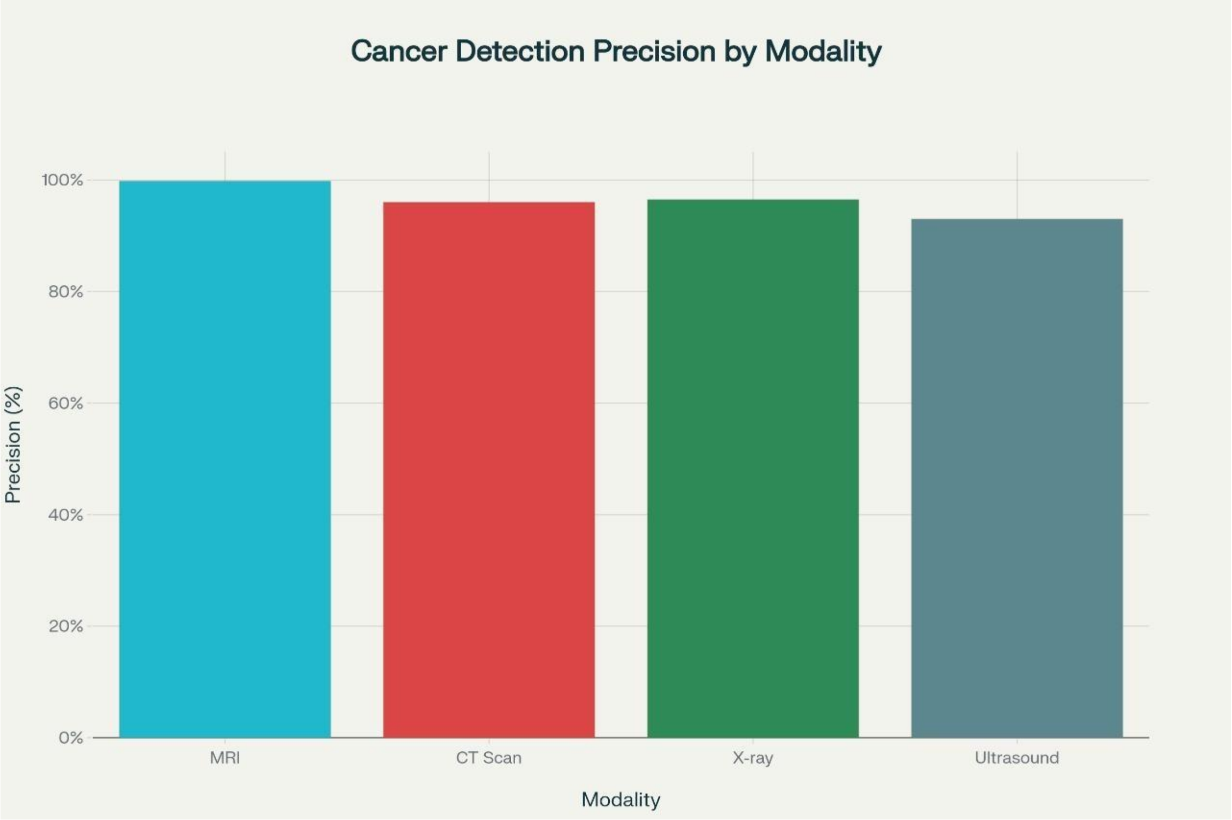


2. Loss



Outputs:





Cancer Detection Precision by Modality
Precision (%)
100%
80%
60%
40%
20%
0%
MRI
CT Scan
X-ray
Ultrasound
Modality