# IMAGE CLASSIFICATION

USAMA SANI (BCB-23S-050)
MUHAMMAD RAYYAN (BCB-23S-046)
MUHAMMAD ANAS (BCB-23S-018)

COURSE : ARTIFICIAL INTELLIGENCE
INSTRUCTOR : MISS AQSA UMER

01
## INTRODUCTION

02
## HOW IT WORKS

03
## CONTRIBUTION

04
## PROJECT CODE

# INTRODUCTION

Image classification is a fundamental task in the field of computer vision, aiming to categorize images into predefined classes or categories. It plays a crucial role in various applications such as medical diagnosis, autonomous driving, facial recognition, and object detection.
At its core, image classification involves the following steps:

**Image Acquisition:** Gathering a dataset of images for analysis.
**Preprocessing:** Preparing the images for analysis, which may include resizing, normalization, and augmentation.
**Feature Extraction:** Identifying and extracting key features or patterns from the images that are indicative of their respective classes.
**Classification:** Using machine learning models, particularly deep learning techniques, to categorize the images based on the extracted features.

# INTRODUCTION

## Why Image Classification is Important

Image classification has revolutionized numerous industries by enabling automation and improving accuracy in tasks that previously relied heavily on human expertise. For example:

**Medical Imaging:** Assists radiologists in identifying diseases from X-rays, MRIs, and CT scans.

**Security:** Enhances surveillance systems through automatic identification of objects and individuals.

**Retail:** Powers visual search engines that allow customers to search for products using images.

## Real-World Applications

The impact of image classification can be seen in various real-world applications:

**Autonomous Vehicles:** Classifies objects on the road to make driving decisions.

**Healthcare:** Detects anomalies in medical images for early diagnosis.

**Social Media:** Automates content moderation by identifying inappropriate images.

# HOW IT WORKS

**Workflow Overview**
The project involves several key steps: data preparation, model development, training, evaluation, and comparison of results.
A flowchart or diagram illustrating this workflow can help visualize the process.
**Data Preparation**
**Dataset:** CIFAR-10 dataset, which includes 60,000 32x32 color images in 10 different classes.
**Preprocessing:**
Load and normalize the images to ensure the pixel values range between 0 and 1.
Split the dataset into training (50,000 images) and testing (10,000 images) sets.
Reshape the labels from (number of samples, 1) to (number of samples,) for compatibility with the models.

# HOW IT WORKS

## Model Development

**Artificial Neural Network (ANN):**

- Architecture: Consists of input, hidden, and output layers.
- Input layer: Flattens the 32x32x3 images into a 1D array.
- Hidden layers: Two dense layers with 5000 and 1000 neurons respectively, using ReLU activation.
- Implementation: Built using a deep learning framework such as TensorFlow & Keras.

**Convolutional Neural Network (CNN):**

- Architecture: Includes convolutional layers, pooling layers, fully connected layers, and the output layer.
- Feature Extraction: Convolutional layers automatically extract features from images.
- Pooling: Reduces dimensionality and retains important information.
- Implementation: Developed using the same framework for consistency.

# HOW IT WORKS

## Training the Models

**Training Process:**
For both models, the training data is fed into the networks.
ANN is trained using Stochastic Gradient Descent (SGD) optimizer.
CNN is trained using Adam optimizer.
Both models use sparse categorical cross-entropy as the loss function.
Training is performed over multiple epochs (5 for ANN and 10 for CNN).

## Model Evaluation

**Evaluation Metrics:**
Accuracy: Measure the percentage of correctly classified images.
Loss: Track the model's performance over epochs.
**Validation:**
Validate the models using the test dataset.
Generate confusion matrices to analyze the performance in each class.
**Classification Report:**
Use sklearn's classification_report to get detailed metrics like precision, recall, and F1-score.

# HOW IT WORKS

## Results Comparison

**Performance Metrics:**

- Compare the accuracy, training time, and loss of ANN and CNN models.
- Visualize the performance through graphs and charts.

**Analysis:**

- Discuss the strengths and weaknesses of each model.
- Highlight the CNN's superior performance in handling image data due to its architecture.

# MEMBERS CONTRIBUTION

**Usama Sani**

**Role:** ANN and CNN Model Development
**Responsibilities:**

- Developed and trained the Artificial Neural Network (ANN) model.
- Developed and trained the Convolutional Neural Network (CNN) model.
- Conducted testing using the CIFAR-10 test Batch.
- Analyzed model predictions to determine accuracy and performance.

.

# MEMBERS CONTRIBUTION

## MUHAMMAD RAYYAN

**Role:** CNN Model Development
**Responsibilities:**

- Developed and trained the Convolutional Neural Network (CNN) model.
- Conducted testing using External images.
- Analyzed model predictions to determine accuracy and performance.

.

# MEMBERS CONTRIBUTION

## MUHAMMAD ANAS

**Role:** ANN Model Development
**Responsibilities:**

- Developed and trained the Convolutional Neural Network (CNN) model.
- Conducted testing using External images.
- Analyzed model predictions to determine accuracy and performance.

.

# PROJECT CODE

## BUILDING SIMPLE ARTIFICIAL NEURAL NETWORK TO TRAIN IMAGES

```
ann = models.Sequential([
    layers.Flatten(input_shape = (32,32,3)),
    layers.Dense(5000,activation = 'relu'),
    layers.Dense(1000,activation = 'relu'),
    layers.Dense(10,activation = 'softmax'),
])
ann.compile( optimizer = 'SGD',
            loss = 'sparse_categorical_crossentropy',
            metrics = ['accuracy'])
ann.fit(x_train,y_train,epochs = 5)
```

# PROJECT CODE

**BUILDING SIMPLE CONVOLUTIONAL NEURAL NETWORK TO TRAIN IMAGES**

```
cnn = models.Sequential([
    layers.Conv2D(filters = 32,kernel_size = (3,3) , activation = 'relu'
    layers.MaxPooling2D((2,2)),
    layers.Conv2D(filters = 64,kernel_size = (3,3) , activation = 'relu'),
    layers.MaxPooling2D((2,2)),
    layers.Flatten(),
    layers.Dense(64,activation='relu'),
    layers.Dense(10,activation='softmax')
])
cnn.compile(optimizer = 'adam',
            loss = 'sparse_categorical_crossentropy' ,
            metrics = ['accuracy'])
cnn.fit(x_train,y_train, epochs = 10)
```

# CONCLUSION

our image classification project utilizing the CIFAR-10 dataset has provided valuable insights into the capabilities of both Artificial Neural Network (ANN) and Convolutional Neural Network (CNN) models. Through meticulous development and testing, we demonstrated the effectiveness of these models in accurately classifying images across ten distinct categories. The ANN excelled in handling structured data, showcasing its versatility and robust performance, while the CNN leveraged its specialized architecture to achieve superior accuracy in image recognition tasks. Each team member played a crucial role: Usama Sani develop of both models and conducted comprehensive testing using the CIFAR-10 dataset, while Rayyan and Anas focused on evaluating the CNN and ANN models, respectively, with external image datasets. Our findings highlight the significance of choosing the right model architecture and testing methodology for optimizing image classification accuracy. Moving forward, this project underscores the potential of neural networks in advancing computer vision applications across various domains.