Deeploy CV Project

Assignment : 3

Name: Harshita Awasthi

Roll Number: 230463

ID : 210 Branch: AE

Date of submission: 25/12/2024

Question 1

The method 1 of the code is based on edge detection and intensity analysis.

1)The flag region detection begins by converting the image to grayscale (image.convert("L")). This simplifies the analysis by reducing the image to a single intensity channel, making it easier to identify patterns like edges.
2)A threshold is applied to create a binary image where pixel values above a certain intensity are set to 255 (white), and others to 0 (black). This binary image enhances contrast and simplifies edge detection.
3)Differences between adjacent pixels in the binary image are calculated using np.diff(). Vertical edges are derived from differences along rows (axis=0), and horizontal edges from differences along columns (axis=1). These differences highlight sharp transitions, which are common at flag boundaries.
4)The calculated vertical and horizontal edges are combined to form an overall edge map. Summing the edge intensities along rows and columns (ysum and xsum) identifies areas where edge densities are highest. The coordinates of these high-density areas are used to define a bounding box.
5)The bounding box is used to crop the grayscale image to the potential flag region.
6)The cropped flag region is resized to a standard size for consistent analysis. The region is split into top and bottom halves, and the mean intensity of each half is calculated.
7)These means are compared to determine the flag.

The second method is based on HSV color space and mask-based analysis.
1)Images are converted from BGR to HSV using cv2.cvtColor(). HSV (Hue, Saturation, Value) separates color information (Hue) from intensity (Value), making it more robust for color-based segmentation.
2)Separate thresholds are defined for red and white in HSV. cv2.inRange() creates binary masks for pixels falling within these thresholds.
3)Masks are smoothed with a Gaussian blur to reduce noise. Canny edge detection (cv2.Canny) is applied to detect edges within the red and white regions.
4)Contours of detected edges are found using cv2.findContours. These contours represent closed curves outlining the red or white regions.
5)For each mask (red and white), the average Y-coordinate of the detected regions is calculated. The relative positions of the red and white regions determine the flag.

Method 1 is based on intensity differences and uses elementary edge detection techniques

for the identification of regions. This is easier but less robust to color variations.
Method 2 relies on color segmentation in HSV space, which makes it more reliable to distinguish red and white flags, especially under changing lighting conditions.

Question 2

a1) In mathematics, the Euclidean distance between two points in Euclidean space is the length of the line segment between them. It can be calculated from the Cartesian coordinates of the points using the Pythagorean theorem, and therefore is occasionally called the Pythagorean distance. the two points can be in any dimension. The formula of calculation of distance is

2) The Mahalanobis distance is a measure of the distance between a point P and a distribution D. Unlike Euclidean distance, which treats all dimensions equally and independently, Mahalanobis distance takes into account the correlations between variables and the scale of the data. It is often used in multivariate analysis to identify outliers or compare points in a non-isotropic space. This distance is zero for P at the mean of D and grows as P moves away from the mean along each principal component axis. If each of these axes is re-scaled to have unit variance, then the Mahalanobis distance corresponds to standard Euclidean distance in the transformed space. The Mahalanobis distance is thus unit-less and scale-invariant and takes into account the correlations of the data set.

3)Manhattan Distance-Manhattan distance is a metric used to determine the distance between two points in a grid-like path. Unlike Euclidean distance, which measures the shortest possible line between two points, Manhattan distance measures the sum of the absolute differences between the coordinates of the points. This method is called "Manhattan distance" because, like a taxi driving through the grid-like streets of Manhattan, it must travel along the grid lines.

Mathematically, the Manhattan distance between two points in an n-dimensional space is the sum of the absolute differences of their Cartesian coordinates. The Manhattan distance formula incorporates the absolute value function, which simply converts any negative differences into positive values. This is crucial for calculating distance, as it ensures all distance measurements are non-negative, reflecting the true scalar distance irrespective of the direction of travel.

b)**ADAM OPTIMIZER-**
Adaptive Moment Estimation is an algorithm for optimization technique for gradient descent. The method is really efficient when working with large problem involving a lot of data or parameters. It requires less memory and is efficient. Intuitively, it is a combination of the 'gradient descent with momentum' algorithm and the 'RMSP' algorithm.
How Adam works?
Adam optimizer involves a combination of two gradient descent methodologies: Momentum: This algorithm is used to accelerate the gradient descent algorithm by taking into consideration the 'exponentially weighted average' of the gradients. Using averages makes the algorithm converge towards the minima in a faster pace.
Root Mean Square Propagation (RMSP): Root mean square prop or RMSprop is an adaptive learning algorithm that tries to improve AdaGrad. Instead of taking the cumulative sum of squared gradients like in AdaGrad, it takes the 'exponential moving average'.

Adam Optimizer inherits the strengths or the positive attributes of the above two methods and builds upon them to give a more optimized gradient descent.

c)**loss function**–

1) The **L2 regularized loss**, also known as **Ridge regression loss**, combines the standard loss function (e.g., mean squared error) with an L2 penalty time to prevent overfitting. This penalty term encourages smaller weights in the model, thereby promoting generalization. the purposes are-

By penalizing large weights, the model is forced to find simpler solutions, reducing variance.

Encourages weights to be small but not exactly zero, unlike L1 regularization.

widely used in regression models.

2) **Ridge regularization** is a type of L2 regularization applied specifically in regression problems, commonly referred to as **Ridge Regression**. It modifies the loss function of linear regression by adding a penalty proportional to the squared magnitude of the model coefficients. This reduces overfitting by discouraging overly complex models with large coefficients.

Prevents Overfitting: Adding the L2 penalty reduces the variance of the model by discouraging large coefficients.

Trade-off Parameter lambda

Small lambda: Ridge regression behaves like standard linear regression. Large lambda: Strong regularization reduces model complexity, potentially leading to under fitting.

Non-Sparse Coefficients: Unlike L1 regularization (Lasso), Ridge does not drive coefficients to exactly zero. Instead, it shrinks all coefficients uniformly.

3) Link to Github Code: Click here.

d) Cross-Entropy Loss

Cross-entropy loss is the measure of the dissimilarity between predicted probabilities and actual labels. This loss function is often applied in classification tasks where it tries to minimize the distance between the predicted probability distribution and the true label distribution.

i) binary-Used for binary classification problems (two classes).

ii) categorical- Used for multi-class classification problems.

e)**Activation Layer** An **activation layer** in neural networks applies a mathematical function to transform the inputs (typically outputs of the previous layer) into a non-linear representation. This transformation is crucial for introducing non-linearities in the network, enabling it to model complex relationships in the data.

The Softmax function is an activation function commonly used in the output layer of classification models. It converts raw scores (logits) into a probability distribution over classes.

Key Properties

Probability Distribution: The outputs are non-negative and sum to 1, representing probabilities.

Amplifies Confidence: Larger logits result in higher probabilities, emphasizing the most likely classes.

Differentiable: Essential for backpropagation and gradient-based optimization.

Applications Used in the output layer of neural networks for multi-class classification tasks.

Combined with cross-entropy loss to train classification models.

The sigmoid function is a commonly used activation function in machine learning and deep learning. It maps any real-valued number to a range between 0 and 1 making it particularly useful for binary classification tasks.
Key Properties
Range: The output of the sigmoid function is always in the range (0,1).
S-Shaped Curve: The sigmoid function is shaped like an "S" (hence the name sigmoid).
Interpretability: Outputs can be interpreted as probabilities, making sigmoid ideal for binary classification.
Smooth and Differentiable: Suitable for gradient-based optimization methods.
Symmetry: The function is centered at 0.5 when x=0.

Tanh introduces non-linearity to the network, maps inputs to the range (1,1) and outputs zero-centered values. This property helps maintain a balanced gradient flow in hidden layers, which can accelerate training. It is primarily used in hidden layers, especially when a model benefits from having outputs distributed symmetrically around zero.
Applications
Hidden Layers: Tanh is often used in hidden layers of neural networks, particularly when zero-centered outputs are desired.
Binary and Multi-Class Classification: Used in certain classification tasks, though sigmoid and softmax are more common in the output layers.
Recurrent Neural Networks (RNNs): Tanh is frequently used in RNNs and LSTMs to process sequential data.

Softmax is the most used for multi-class classification as it produces probabilities that add up to 1. Sigmoid is used extensively in binary classification for producing probabilities. Tanh is less popular because of vanishing gradient but may be used in hidden layers where zero-centered outputs are needed.

f) The learning rate is a hyperparameter in machine learning that controls the size of the step taken during each update of the model's parameters (weights and biases) to minimize the loss function. It determines how quickly or slowly a model learns.
Small Learning Rate: Leads to slow convergence but provides more precise updates. Risk: Training can get stuck in local minima or take too long.
Large Learning Rate:Faster convergence but may overshoot the optimal solution or cause divergence.
Risk: Model may oscillate around or miss the optimal parameters. Optimal Learning Rate:
Balances convergence speed and stability. Often determined using techniques like learning rate schedules, adaptive optimizers (e.g., Adam), or trial and error.
Learning Rate Schedules
Constant: Fixed value throughout training.
Decay: Gradually decreases over epochs to refine learning.
Cyclical: Varies periodically to escape local minima and improve exploration

g)Batches refer to subsets of the dataset used during training or inference in machine learning models. Instead of processing the entire dataset at once (which can be compu-

tationally expensive), data is divided into smaller chunks called batches.
Why Use Batches?
Efficiency: Reduces memory requirements by processing a portion of data at a time.
Stochastic Optimization: Allows models to learn faster by updating weights after each batch rather than after the entire dataset.
Parallelism: Batches enable parallel processing, speeding up computations.

Batch Dimension: When inputing data into a model, the first dimension typically represents the batch size. For example: An image array of shape (height,width,channels) becomes (batchsize,height,width,channels).
example of the code
import numpy as np
image = np.random.rand(224, 224, 3) Example: Single image
batchimage = np.expand$_d ims(image, axis = 0) Shapebecomes(1, 224, 224, 3)$

h) Gradient Descent is an optimization algorithm used to minimize a loss function by iteratively adjusting the model's parameters (weights and biases). Calculates the gradient (partial derivative) of the loss function with respect to the parameters and updates them in the opposite direction of the gradient to reduce the loss.

i) The table details the proper loss function and activation function to apply for the specific classification task type:
Table Interpretation 1 or 2 Classes (Binary Classification):
Class Mode: Binary
Loss Function: Binary Cross-Entropy
It computes the error of the true label and predicted probabilities in binary outcomes.
Activation Function (Last Layer): Sigmoid
This function transforms the output into a range of 0 up to 1, that is probabilities Multiclass, Single Label (Multiclass Classification):
Class Mode: Categorical Loss Function: Categorical Cross-Entropy Calculates the error for multi-class predictions for which only one label is accurate. Activation Function (Output Layer): Softmax Transforms logits into the probabilities that sum up to 1 1, applied in multi-class classification. Multi-class, Multilabel (Multi-label Classification) Class Mode: Categorical Loss Function: Binary Cross-Entropy
Calculation of errors is handled independently for classes.
Activation Function (Output Layer): Sigmoid
Independent labels can be right, it deals with individual probabilities per label.
Important Points
The loss function measures errors based on type of classifications. The activation function ensures the model outputs are in a range suitable for interpretation, for example, probabilities. In the case of multi-label tasks, sigmoid is used to handle independent probabilities for each label.
k) In machine learning, a **basis function** is a mathematical function used to transform input data into a new space, facilitating the modeling of complex, non-linear relationships through linear combinations of these transformed inputs. This approach enables linear models to capture intricate patterns in the data by applying non-linear transformations to the input features.

Question 3

l) The founders of OpenAi, Alex Krizhevsky, Geoffrey Hinton and Ilya Sutskever co-authored the groundbreaking 2012 research paper, 'AlexNet'. This convolutional neural network significantly advanced the field of deep learning by achieving unprecedented results in image recognition tasks.

Question 5

Link to Github Code: Click here.