# Explanation of Plots:

A screenshot of a computer program

Description automatically generated

Proposed caption for the plot: *Aggregate heatmap of pixel importance generated by summing LIME explanation masks across all images in the dataset. Brighter regions indicate pixels that were consistently deemed important for the model's predictions, while darker regions represent areas of lesser significance.*

The aggregated heatmap of pixel importance visualizes the regions in the input images that the LIME (Local Interpretable Model-agnostic Explanations) explainer identified as the most important for the model's predictions. Each pixel in the heatmap corresponds to how frequently and strongly that pixel contributes to the model's decision across the entire dataset.

Methodology used for the heatmap:  
  
Objective: To identify and aggregate the regions of interest across all the images in the dataset where the model focuses its attention during classification.

Steps:

* **Generate Individual Masks**: For each image, the LIME explainer generates a **binary mask** (mask), which highlights the pixels that contribute the most to the predicted class.
* **Aggregate Masks**: The masks for all images are summed together, producing a **cumulative heatmap** that reflects the importance of each pixel across the dataset.
* **Visualization**: The heatmap is visualized using a **color gradient** where:
  + Darker regions indicate lower importance or infrequent use in explanations.
  + Brighter regions (yellow) indicate high importance and frequent contribution to predictions.

Interpretation of the Heatmap:

1. **Key Regions**:
   * The bright regions in the heatmap show the key areas of interest for the model. These areas were consistently marked as significant by the LIME explainer across multiple images.
   * The concentration of pixel importance in specific regions suggests that the model relies heavily on these areas for classification.
2. **Dark Regions**:
   * The dark regions in the heatmap represent pixels that are rarely or never used in the model's decision-making process.
3. **Global Insights**:
   * The heatmap provides a dataset-wide explanation of the model's behaviour, going beyond single-instance explanations to reveal patterns in how the model processes data.
   * This is useful for understanding whether the model relies on meaningful, interpretable features or possibly irrelevant or spurious regions.

Insights from the Aggregate areas of Pixel Importance:

* This visualization serves as evidence that the model captures **relevant spatial features** in the images, which can validate the quality of the dataset and the effectiveness of the model architecture.
* The heatmap enhances interpretability by showing which regions of the images influence the model's decisions. This can build trust in the model's predictions.

A graph of a graph

Description automatically generated

Proposed caption for the plot: *Average confidence scores for predictions across the three classes. The model exhibits the highest confidence for Class 1 (~0.6), while Classes 0 and 2 show lower confidence (~0.3 and ~0.2, respectively). This analysis highlights variations in class-specific feature learnability and potential dataset or model biases.*

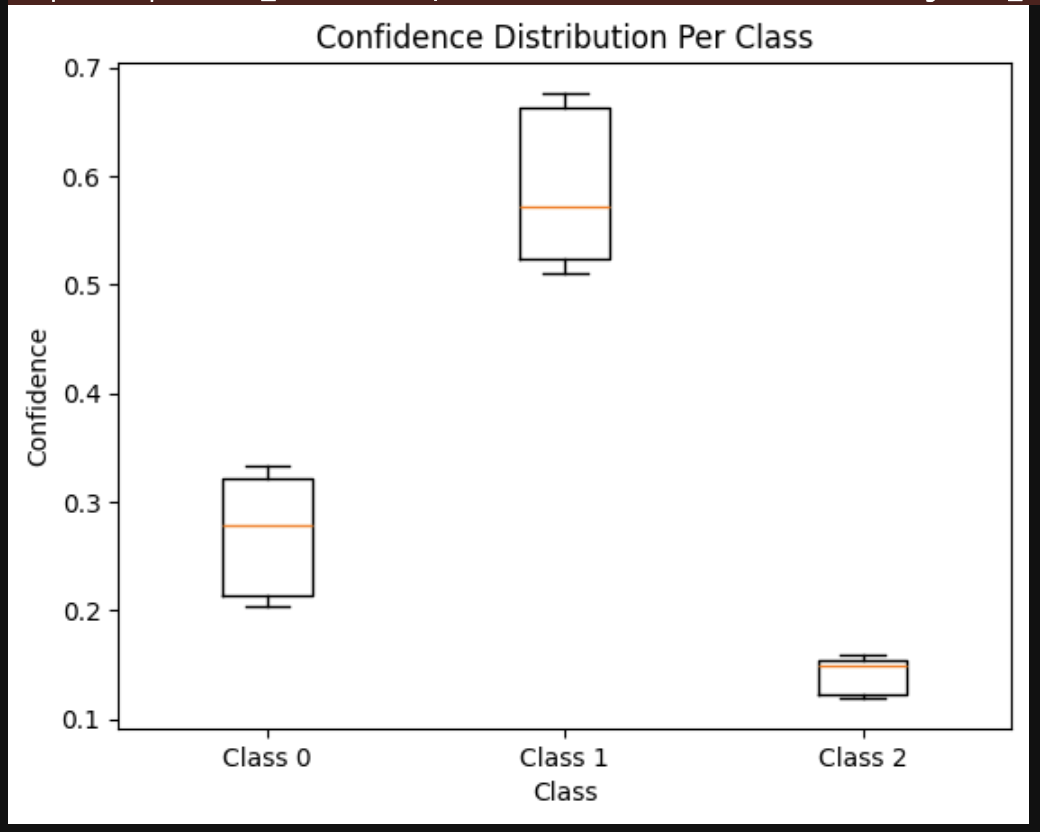
The bar plot displays the **average confidence** of the model’s predictions for each class. Confidence is the probability value assigned by the model's **output layer** to its predicted class. The higher the confidence for a class, the more certain the model is about its predictions for that class.

Key Observations:

1. Class-Wise Confidence:
   * The bar heights represent the average confidence for each class.
   * The class labels (0, 1, 2) correspond to the three distinct categories in the dataset:
     + Class 0: Moderate average confidence (~0.3).
     + Class 1: High average confidence (~0.6).
     + Class 2: Low average confidence (~0.2).
2. Dominance of Class 1:
   * The model exhibits the highest confidence for Class 1, suggesting that the features associated with this class are more distinct or easier for the model to identify.
3. Lower Confidence for Classes 0 and 2:
   * For Class 0 and Class 2, the model shows relatively lower average confidence, which may indicate:
     + Overlap in features between these classes.
     + Lack of sufficient distinguishing features.
     + Potential bias or imbalance in the dataset.

Significance of the plot:

1. Model Behaviour:
   * This analysis reveals how confidently the model makes predictions across different classes, providing insights into class-specific performance.
   * High confidence for Class 1 indicates that the model has learned meaningful features for this class, while lower confidence for Classes 0 and 2 warrants further investigation.
2. Dataset Quality:
   * The confidence distribution could reflect the quality of the dataset:
     + Classes with low confidence may suffer from noise, insufficient data, or overlapping features.
3. Bias Detection:
   * If one class consistently shows higher confidence compared to others, this could suggest bias in the model or dataset.
4. Potential Improvements:
   * Collect more data for classes with low confidence.
   * Apply data augmentation to enrich the feature space for underperforming classes.
   * Examine feature importance to identify potential overlap between classes.



Plot caption: *Box plot showing the distribution of confidence scores for each class.*

Explanation:

* Highlights the spread of confidence scores for each class, including:
  + Median (orange line).
  + Interquartile range (box edges).
  + Minimum and maximum (whiskers).
  + Outliers (if any).
* Provides a detailed view of how the model's confidence varies within each class.
* Class 1:
  + Highest confidence scores overall.
  + Small spread (most scores are tightly clustered around the median).
* Class 0:
  + Moderate confidence scores with a larger spread, indicating variability in predictions.
* Class 2:
  + Lowest confidence scores overall, with a very narrow range, suggesting low certainty across predictions for this class.

# Explanation of the Dataset and the CNN:

A black and white label

Description automatically generated A white circle with black background

Description automatically generated A black and white square with a black background

Description automatically generated

Proposed caption*: Representative images from each class in the dataset: (1) No Phase Shift (Label 0) with horizontally symmetrical patterns, (2) π/2 Phase Shift (Label 1) showing circular symmetry, and (3) Zero Phase Shift (Label 2) characterized by dense and irregular patterns.*

The dataset consists of grayscale images representing different phase shifts categorized into three distinct classes:

1. **No Phase Shift (Label 0)**:
   * Images in this class exhibit a **horizontal symmetrical pattern**, with clear parallel bands or regions.
   * Uniform regions where no phase shift occurs, resulting in a stable and consistent spatial distribution.
2. **π/2 Phase Shift (Label 1)**:
   * Images in this class show **rotational symmetry** with distinct, circular features.
   * The patterns suggest regions of strong, periodic phase changes that create localized zones of influence, leading to a symmetrical and structured distribution.
   * This class is visually distinct from the other two classes due to its circular symmetry and isolated bright regions.
3. **Zero Phase Shift (Label 2)**:
   * Images in this class feature **dense, irregular patterns**, suggesting higher complexity or variability in the phase distribution.
   * Unlike the other classes, this category exhibits **smaller, dispersed regions of importance** interspersed across the image, with no clear global symmetry.
   * This class appears noisier and more intricate, which may challenge the model’s ability to generalize effectively.

Insights on Image Characteristic

1. Class-Specific Features:
   * Each class exhibits distinct patterns:
     + Class 0: Linear and horizontally aligned regions.
     + Class 1: Circular and symmetrical features.
     + Class 2: Dense, irregular patterns.
2. Separation Between Classes:
   * The distinct geometric features (lines, circles, and irregular distributions) allow for meaningful separation between classes.
   * However, the higher variability in Class 2 might make it harder for the model to identify key features, as suggested by its lower confidence in the model's predictions.
3. Inter-Class Differences:
   * Class 1 has the most visually distinct pattern, making it easier for the model to classify.
   * Class 0 and Class 2 might have overlapping or less distinct features, increasing the likelihood of misclassification.
4. Challenges:
   * Class Imbalance: If one class dominates the dataset, the model might favor it during predictions.
   * Complexity in Class 2: The irregular patterns could lead to lower confidence scores and greater prediction variability.

Neural Network Architecture:

A screenshot of a computer

Description automatically generated

**Input Layer**

* Accepts images with dimensions (256, 256, 1) as input, representing grayscale images.

**Feature Extraction Layers**

1. **Convolutional Block 1**:
   * **Conv2D Layer**:
     + Applies 32 filters of size (3, 3) with ReLU activation.
     + L2 regularization (l2=0.01) is used to mitigate overfitting.
     + Output Shape: (254, 254, 32) (after convolution).
   * **Batch Normalization**:
     + Normalizes activations to stabilize training.
   * **MaxPooling2D**:
     + Reduces spatial dimensions by a factor of 2.
     + Output Shape: (127, 127, 32).
   * **SpatialDropout2D**:
     + Randomly deactivates 30% of features for better generalization.
2. **Convolutional Block 2**:
   * **Conv2D Layer**:
     + Applies 64 filters of size (3, 3) with ReLU activation.
     + Output Shape: (125, 125, 64).
   * **Batch Normalization**:
     + Normalizes activations to stabilize training.
   * **MaxPooling2D**:
     + Reduces spatial dimensions by a factor of 2.
     + Output Shape: (62, 62, 64).
3. **Convolutional Block 3**:
   * **Conv2D Layer**:
     + Applies 128 filters of size (3, 3) with ReLU activation.
     + Output Shape: (60, 60, 128).
   * **Batch Normalization**:
     + Normalizes activations to stabilize training.
   * **MaxPooling2D**:
     + Reduces spatial dimensions by a factor of 2.
     + Output Shape: (30, 30, 128).

**Global Average Pooling**

* **GlobalAveragePooling2D**:
  + Aggregates spatial information into a single vector per channel.
  + Reduces the output to a feature vector of size (128).

**Fully Connected Layers**

1. **Dense Layer**:
   * 256 units with ReLU activation.
   * L2 regularization (l2=0.01) applied for better generalization.
   * Output Shape: (256).
2. **Dropout**:
   * Drops 50% of the neurons to prevent overfitting.
3. **Output Layer**:
   * 3 units with softmax activation, corresponding to the three output classes.
   * Produces class probabilities for multi-class classification.

**Regularization Techniques**

* **L2 Regularization**:
  + Applied in all Conv2D and Dense layers to reduce overfitting by penalizing large weights.
* **Batch Normalization**:
  + Speeds up convergence and stabilizes training by normalizing layer outputs.
* **Spatial Dropout and Dropout**:
  + SpatialDropout2D deactivates feature maps, while Dropout deactivates neurons, both preventing over-reliance on specific features.

**Parameters**

* **Total Parameters**: 127,363.
  + **Trainable Parameters**: 126,915 (weights and biases updated during training).
  + **Non-Trainable Parameters**: 448 (parameters from BatchNormalization layers).

**Key Characteristics**

1. **Efficiency**:
   * The model is lightweight, with fewer than 130K parameters, making it suitable for environments with limited computational resources.
2. **Modularity**:
   * Uses three distinct convolutional blocks for hierarchical feature extraction.
3. **Scalability**:
   * Can be scaled for larger datasets or higher resolutions by increasing filter sizes or the number of layers.

* Show graphs for train and test accuracy (for phase shifters and multiplexers)
* Any other insightful graphs that we can generate?? – check