**Hyperparameter Exploration**

**Hyperparameters and Results:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Epochs** | **Kernel Size** | **Learning Rate** | **Channel** | **Class Weights** | **Loss** | **Accuracy** |
| 1 | 12 | 3 | 0.000001 | 1 | Yes | N/A | 87.5% |
| 2 | 8 | 3 | 0.000001 | 1 | Yes | 0.3295 | 75.0% |
| 3 | 12 | 5 | 0.000001 | 1 | Yes | 0.2744 | 87.5% |
| 4 | 12 | 3 | 0.001 | 1 | Yes | 0.274 | 87.5% |
| 5 | 12 | 3 | 0.000001 | 1 | No (Without Class Weights) | 0.334 | 87.5% |
| 6 | 12 | 3 | 0.000001 | 1 | No (With SGD Optimizer) | 0.1969 | 93.75% |

**Optimizer Configuration:**

sgd\_optimizer = SGD(learning\_rate=0.01, momentum=0.9, nesterov=True)

model.compile(optimizer=sgd\_optimizer, loss='binary\_crossentropy', metrics=['accuracy'])

**Observations and Insights**

**Observations:**

1. **Learning Rate:**
   * A low learning rate (0.000001) requires more epochs to achieve good accuracy (75% at 8 epochs vs. 87.5% at 12 epochs).
   * Higher learning rates (e.g., 0.001) enable faster convergence, achieving similar accuracy in fewer epochs.
2. **Kernel Size:**
   * Changing the kernel size from 3 to 5 does not significantly affect accuracy (87.5% in both cases). Larger kernels may better capture spatial patterns depending on the dataset.
3. **Number of Epochs:**
   * Increasing epochs improves accuracy by allowing the model to learn more effectively (75% at 8 epochs vs. 87.5% at 12 epochs).
4. **Class Weights:**
   * Accuracy remains consistent at 87.5% with and without class weights, suggesting class imbalance was not a significant issue.
5. **Optimizer:**
   * The SGD optimizer with momentum and Nesterov acceleration improves accuracy to 93.75%, demonstrating the benefits of advanced optimization techniques.
6. **Loss Behavior:**
   * Lower loss values correlate with higher accuracy (e.g., loss = 0.1969 and accuracy = 93.75%).

**Reasons Behind Observations:**

1. **Learning Rate:**
   * Lower learning rates slow down convergence, necessitating more epochs for good performance. Higher rates accelerate learning but may risk overshooting or instability.
2. **Kernel Size:**
   * Smaller kernels capture finer details, while larger kernels identify broader patterns. The similar accuracy suggests that the dataset did not significantly benefit from larger kernels.
3. **Number of Epochs:**
   * More epochs allow better pattern learning. However, excessive training risks overfitting.
4. **Class Weights:**
   * Class weights are effective in addressing data imbalances by penalizing misclassification of underrepresented classes. Their lack of impact here indicates balanced class representation.
5. **Optimizer:**
   * Momentum and Nesterov acceleration improve training by using prior gradients and reducing oscillations, resulting in faster and more stable convergence.
6. **Loss and Accuracy:**
   * Reduced loss values signify improved predictions, aligning with increased accuracy.

**Conclusion:**

The results highlight the impact of hyperparameter tuning and optimization techniques on model performance. By balancing learning rate, kernel size, and the number of epochs and leveraging advanced optimizers, significant improvements in accuracy can be achieved.

**Accuracy vs Loss Graphs:**

1.A graph of different colored lines

Description automatically generated with medium confidence

2.A graph of different colored lines

Description automatically generated with medium confidence

3.A graph of a line and a line

Description automatically generated with medium confidence

4.A graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of a graph of

Description automatically generated

5.A graph of different colored lines

Description automatically generated with medium confidence

6.

A screenshot of a graph

Description automatically generated