Giada Griso
September 24, 2024
Machine Learning Course
Exercise 2.5

Visual Applications of Machine Learning

Part 1 – Handwriting Recognition

```python
# Calculate accuracy

from sklearn.metrics import accuracy_score

accuracy = accuracy_score(y_real_test, predicted_labels)
print(f"Model accuracy on handwritten data: {accuracy * 100:.2f}%")
```

```
Model accuracy on handwritten data: 50.00%
```

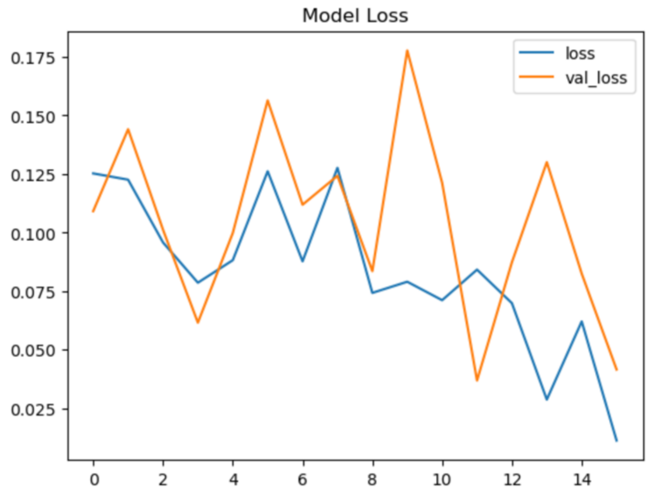
```python
# Display the predicted and true labels

for i, (pred, actual) in enumerate(zip(predicted_labels, y_real_test)):
    print(f"Image {i}: Predicted = {pred}, Actual = {actual}")
```

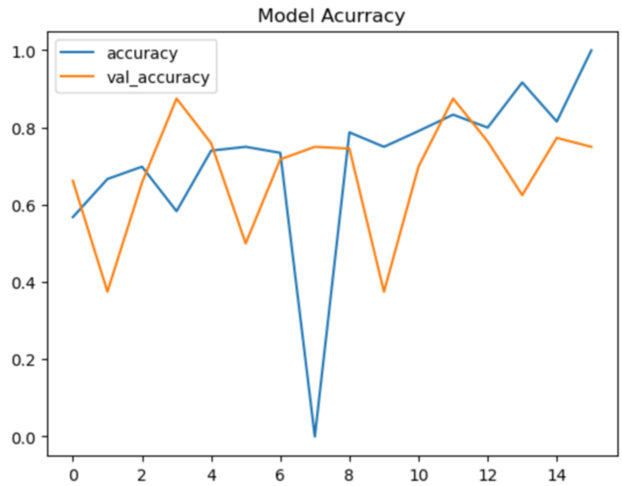
```
Image 0: Predicted = 0, Actual = 0
Image 1: Predicted = 4, Actual = 1
Image 2: Predicted = 2, Actual = 2
Image 3: Predicted = 2, Actual = 3
Image 4: Predicted = 4, Actual = 4
Image 5: Predicted = 5, Actual = 5
Image 6: Predicted = 6, Actual = 6
Image 7: Predicted = 4, Actual = 7
Image 8: Predicted = 2, Actual = 8
Image 9: Predicted = 8, Actual = 9
```

Part 2 – Radar Recognition

Using 16 epochs, the results are the following:

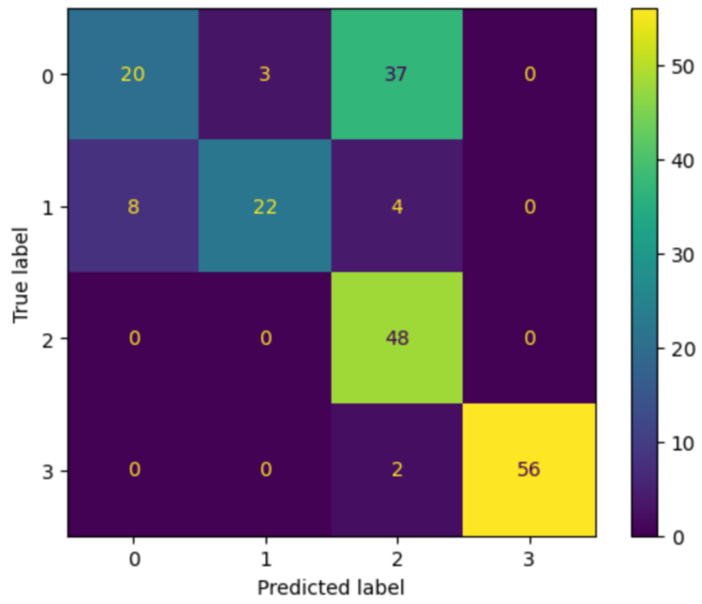
Accuracy	1,0
Val_accuracy	**0,75**
Loss	0,011275663040578365
Val_loss	0,041568450629711115





High validation accuracy suggests that the model is performing well on data outside of the training set. The gap between training and validation accuracy suggests the model may be overfitting to the training data, as it performs perfectly on data it has seen but struggles a bit on unseen data. The loss on the training dataset is very low, indicating that the model has minimized the error on the training set.

The loss on the validation dataset is higher than the training loss. This further indicates a potential overfitting issue, where the model fits the training data very well (low loss and high accuracy) but generalizes poorly to unseen data (higher validation loss and lower validation accuracy).



In the confusion matrix, we can observe that **cloudy** and **rain** images have relatively more misclassifications compared to the other two classes. Especially, "cloudy" is frequently misclassified as "shine" (37 times). **Shine** and **sunrise** images are classified with high accuracy, in fact there are very few misclassifications for these classes. The confusion matrix highlights that the model struggles the most with distinguishing between "cloudy" and "shine", as well as between "cloudy" and "rain".

Make a proposal for the use of GANs in weather prediction:

- Using sky satellite images of a certain area, GANs could track storms or precipitation patterns.
- GANs could be trained on historical weather data and then used to simulate future climate scenarios, helping predict how extreme weather events might become more frequent or severe in certain regions.
- GANs could be trained to process satellite or radar imagery and automatically segment and label different meteorological phenomena (like storms and hurricanes).

Apparently, there have been several studies on the applications of GANs in weather prediction:

https://www.nature.com/articles/s41598-023-35093-9

https://s3.us-east-1.amazonaws.com/climate-change-ai/papers/neurips2019/21/paper.pdf

https://www.preventionweb.net/media/82579/download?startDownload=20240924