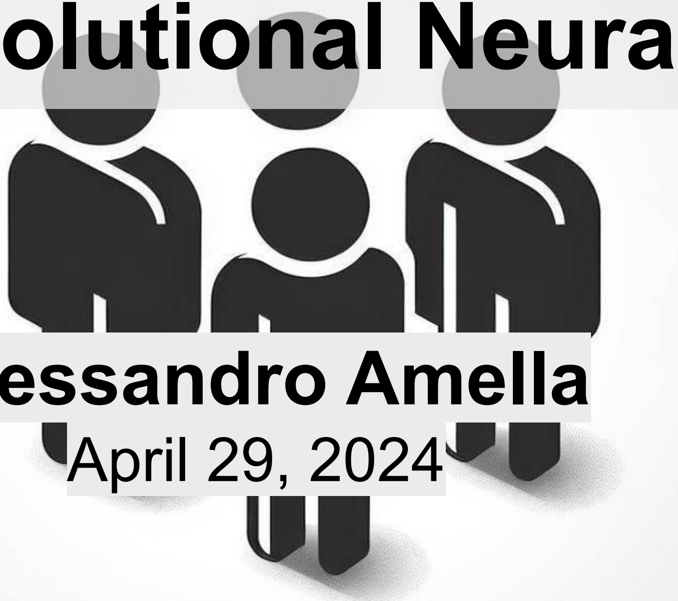
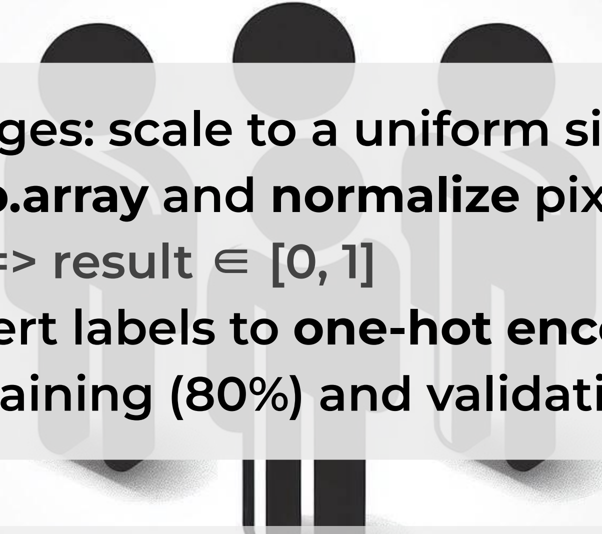
Classification of Room Occupancy using a Convolutional Neural Network



Alessandro Amella

April 29, 2024

Preprocessing

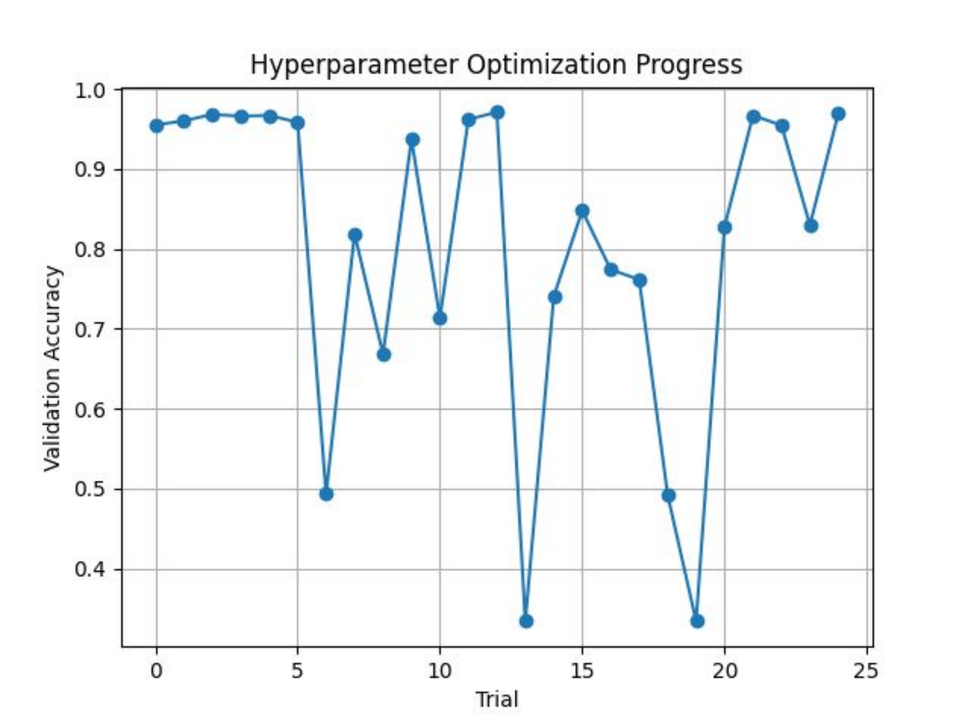


- **Crop** and **resize** images: scale to a uniform size
- Convert image to **np.array** and **normalize** pixel values
  - pixel val / 255.0 => result ∈ [0, 1]
- Encode labels: convert labels to **one-hot encoding**
- Splitting data into training (80%) and validation (20%) sets

Hyperparameter Optimization

Utilized **Optuna** to optimize:

- **conv2d_filter**: number of filters in convolutional layers
- **dense_units**: number of neurons in a dense layer
- **dropout_rate**: rate for dropout regularization
- **learning_rate**: learning rate for Adam optimizer



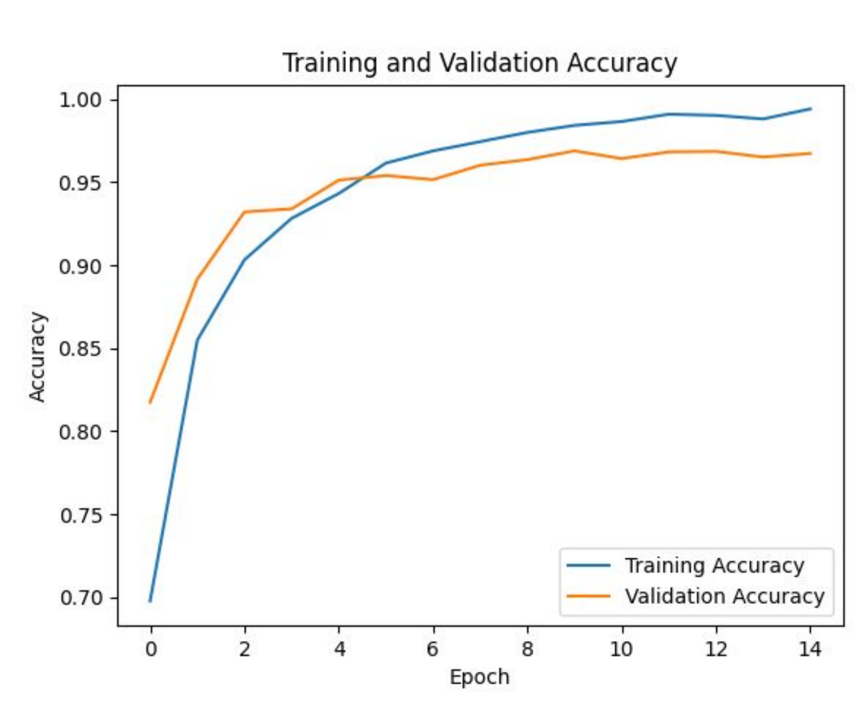
The final CNN Model

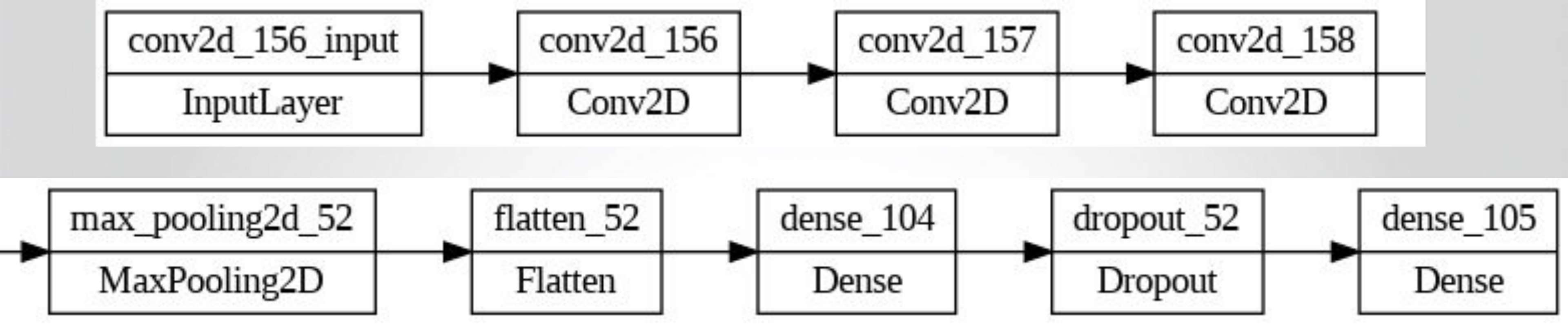
Used **CNN** for classification with:

- **Adam** compiler
- **Categorical Cross-Entropy** loss function

To limit **overfitting:**

- **Early stopping**
- **Keras Pruning:** prunes unpromising trials





- Input Layer: expects color images with dimensions of 44x50 px
- **3 Conv2D** layers for feature extraction using *best_conv2d_filter*
- **MaxPooling2D** layer to shrink feature maps
- **Flatten** layer to flatten output to 1D vector
- **Dense** layer (fully connected) with *best_dense_units* neurons
- **Dropout** rate with rate of *best_dropout_rate*
- Output **Dense** layer of 4 neurons for 4 output classes

Image Augmentation

Explored image augmentation techniques

- Rotation, flipping, scaling
- Low accuracy on validation data, even with non-aggressive parameters

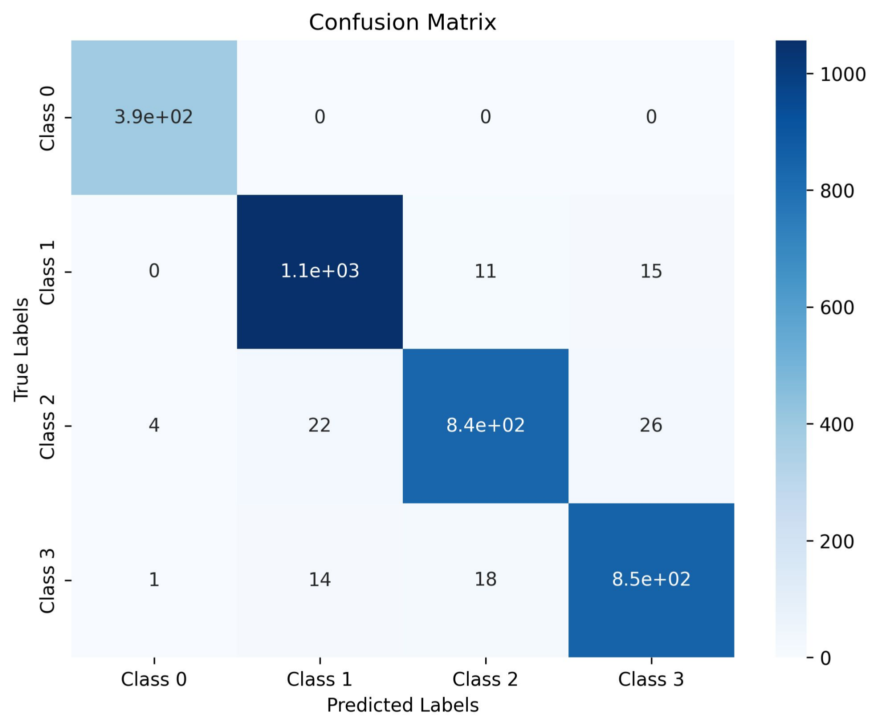
=> **Overfitting**

=> Original dataset already **sufficiently diverse**

Results and Conclusion

After 50 epochs, the CNN model achieved a validation accuracy of 96.94% 🎉🎉

=> Good hyperparameter optimization by Optuna, good choice of architecture


Confusion Matrix
True Labels
Predicted Labels
Class 0
Class 1
Class 2
Class 3
3.9e+02
1.1e+03
8.4e+02
8.5e+02