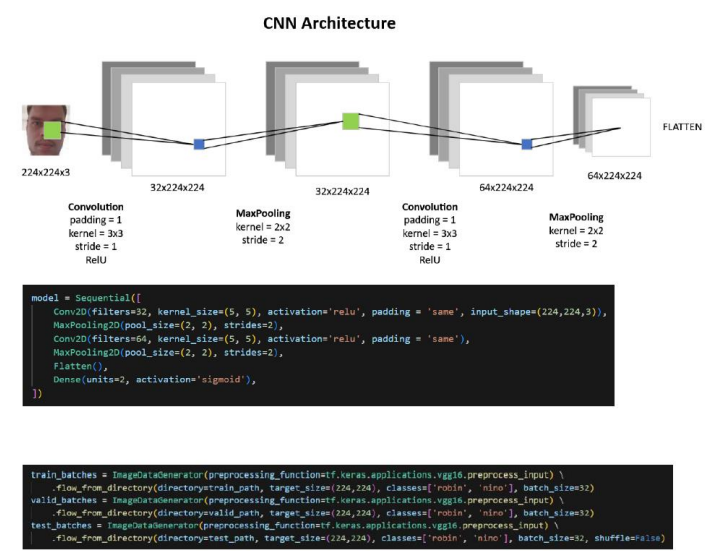


Project: Face Recognition using DL



- Face Detection with CNN
- Datensammlung
 - Klassen: 0, 1
 - Detect Face in Bild
 - Resize & Reshape (224x224x3)
 - Augmentation
 - Preprocessing / Labeling

⇒ Input für das zu trainierende Modell

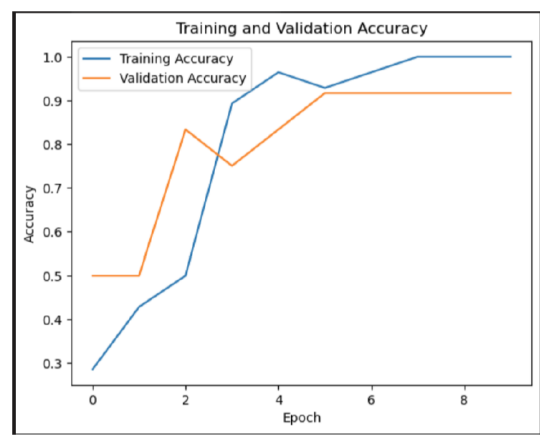
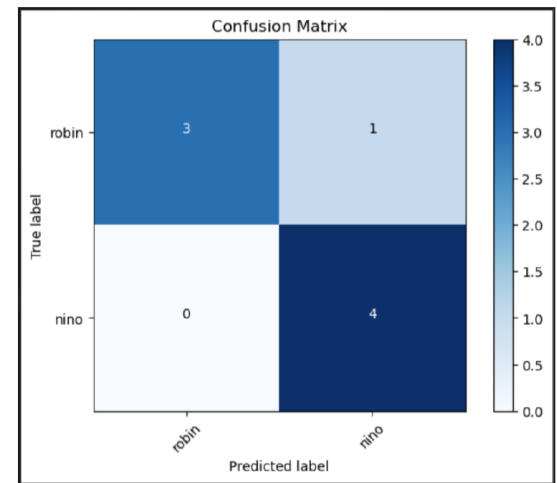


Eigenes Modell

Preprocessing der Bilder mit VGG16 methode

Acc = 95% nach bereits 4 Epochen

Diverse Änderungen am Modell vorgenommen während entwicklung



```
model.compile(loss='categorical_crossentropy', optimizer = 'adam', metrics=["accuracy"])
history = model.fit(x=train_batches, validation_data=valid_batches, epochs=10, verbose=2)

# Plot the learning curve
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.title('Training and Validation Accuracy')
plt.legend()
plt.show()
```

Umfang Datensammlung

Train Set (inkl. Augmented):
392

Test Set:
14

Validation Set:
21

- ✓ face_dataset
- ✓ test
 - > nino
 - > robin
- ✓ train
 - > nino
 - > robin
- ✓ valid
 - > nino
 - > robin

Face Regognition App mit StreamLit

Erstelltes Modell gespeichert und wird per Python Script geladen, damit «Live» Aufnahmen Klassifiziert werden können

Face Recognition App

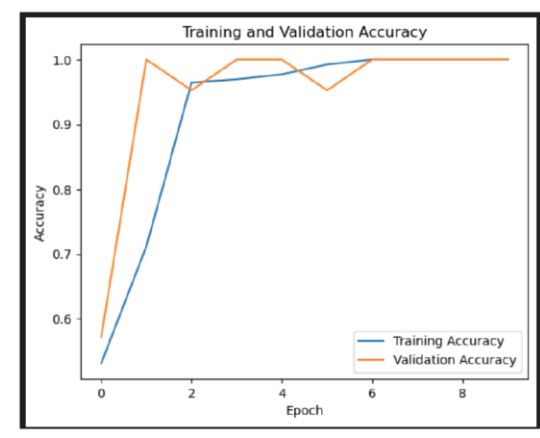
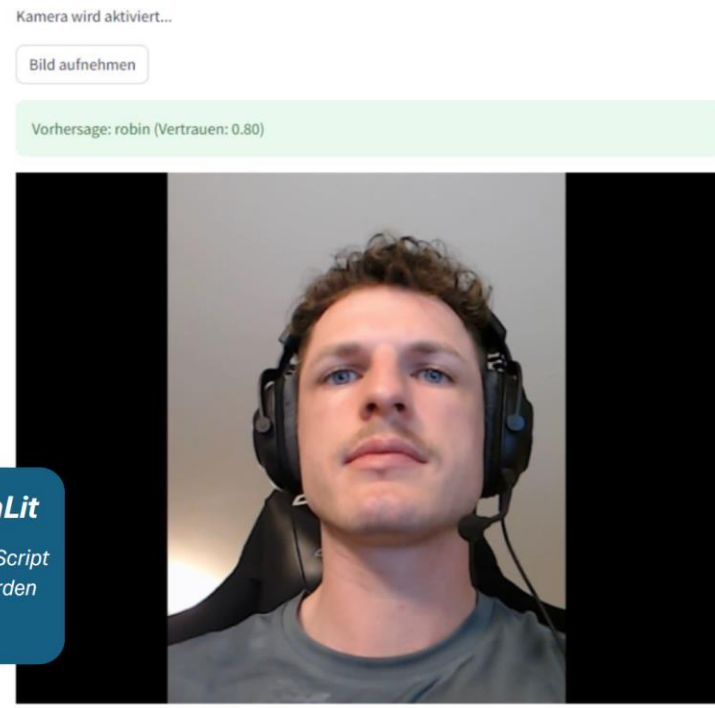


Image 1: Actual Class = robin, Predicted Class = robin, Confidence = 1.00
Image 2: Actual Class = robin, Predicted Class = robin, Confidence = 1.00
Image 3: Actual Class = robin, Predicted Class = robin, Confidence = 1.00
Image 4: Actual Class = robin, Predicted Class = robin, Confidence = 1.00
Image 5: Actual Class = robin, Predicted Class = robin, Confidence = 1.00
Image 6: Actual Class = robin, Predicted Class = robin, Confidence = 1.00
Image 7: Actual Class = robin, Predicted Class = robin, Confidence = 1.00
Image 8: Actual Class = nino, Predicted Class = nino, Confidence = 1.00
Image 9: Actual Class = nino, Predicted Class = nino, Confidence = 1.00
Image 10: Actual Class = nino, Predicted Class = nino, Confidence = 1.00
Image 11: Actual Class = nino, Predicted Class = nino, Confidence = 1.00
Image 12: Actual Class = nino, Predicted Class = nino, Confidence = 1.00
Image 13: Actual Class = nino, Predicted Class = nino, Confidence = 1.00
Image 14: Actual Class = nino, Predicted Class = nino, Confidence = 1.00

Pre-Trained Modell von Keras

VGG Modell

Preprocessing der Bilder mit VGG16 methode

Bereits nach 2 Epochen bei Acc = 95%

