FIAT LUX

Bewegungsmelde - Aktivierung

Features

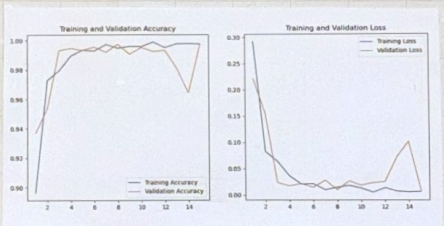
3700 images

PERSON

Models

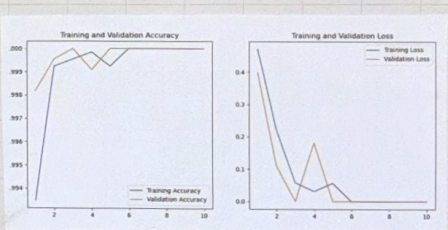
Selfmade

```
model = Sequential()
model.add(Convolution2D(16, kernel_size=(3,3),padding='same', activation='relu',input_shape=(224,224,3)))
model.add(Convolution2D(16, kernel_size=(3,3),padding='same', activation='relu'))
model.add(MaxPooling2D((2,2)))
model.add(Convolution2D(16,kernel_size=(3,3),padding='same', activation='relu'))
model.add(Convolution2D(16,kernel_size=(3,3),padding='same', activation='relu'))
model.add(MaxPooling2D((2,2)))
model.add(Flatten())
model.add(Dense( units 100, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))
model.add(Dense( units 100, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))
model.add(Dense( units 1, activation='sigmoid'))

# compile model and initialize weights
model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

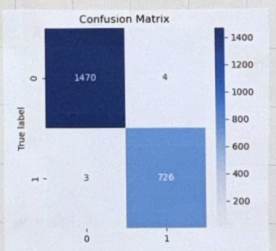


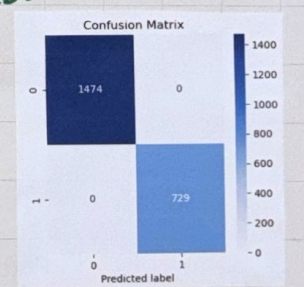
VGG16

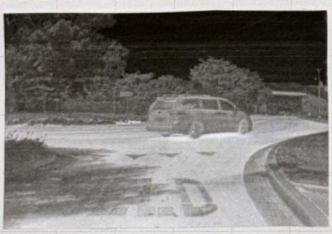
```
# Load the VGG16 model pre-trained on ImageNet data, excluding the top layer
base_model = VGG16(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

# Freeze the layers of the base model
base_model.trainable = False

# Add custom layers on top of the base model
x = base_model.output
x = Flatten()(x)
x = Dense( units 1024, activation='relu')(x)
predictions = Dense( units 1, activation='sigmoid')(x)  # sigmoid for binary classific

# Define the full model
model = Model(inputs=base_model.input, outputs=predictions)

# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```



Not-cropped





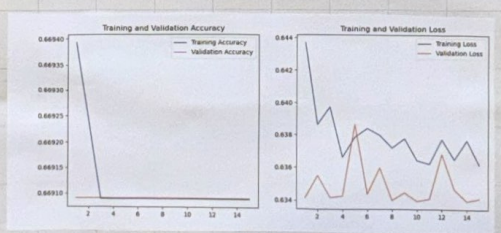


3700 images

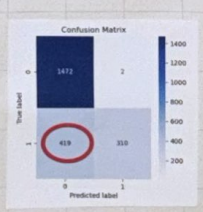
3700 images

Non-Person

- Annotations (cropping/size)
- Balancing



Sigmoid



cropped tested with full images

P: 0.99
R: 0.43
F1: 0.57