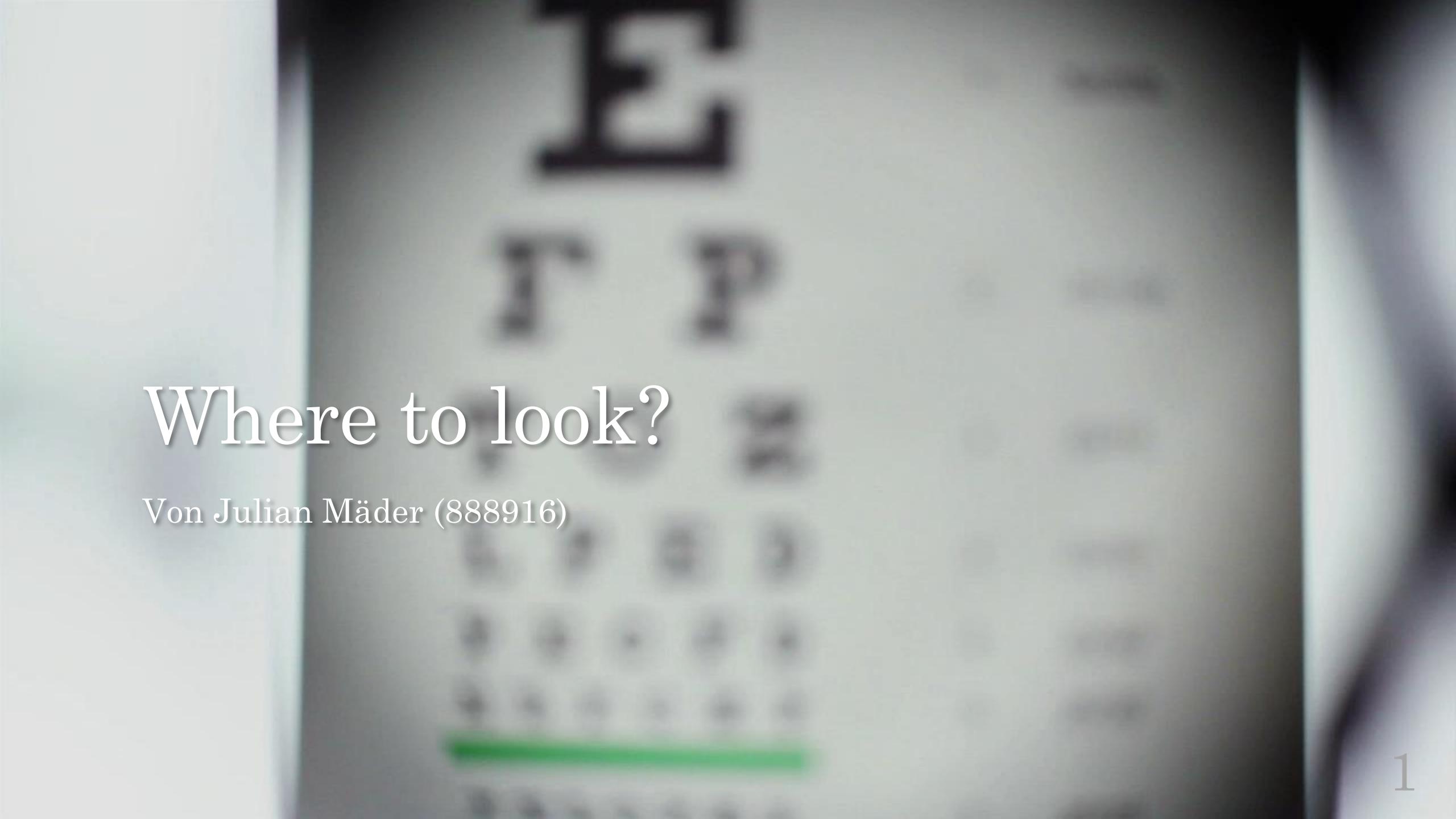


Where to look?

Von Julian Mäder (888916)

Inhalt

- Einleitung
- Class activision maps (CAM)
- Implementierung von CAMs
- Was kann das Programm/ wo liegen Limitierungen?
- Programm Struktur
- Ergebnisse

Einleitung

Woran macht ein Neuronales Netz seine Entscheidung fest?

Einleitung

Woran macht ein Neuronales Netz seine Entscheidung fest?

Meistens wissen wir das gar nicht

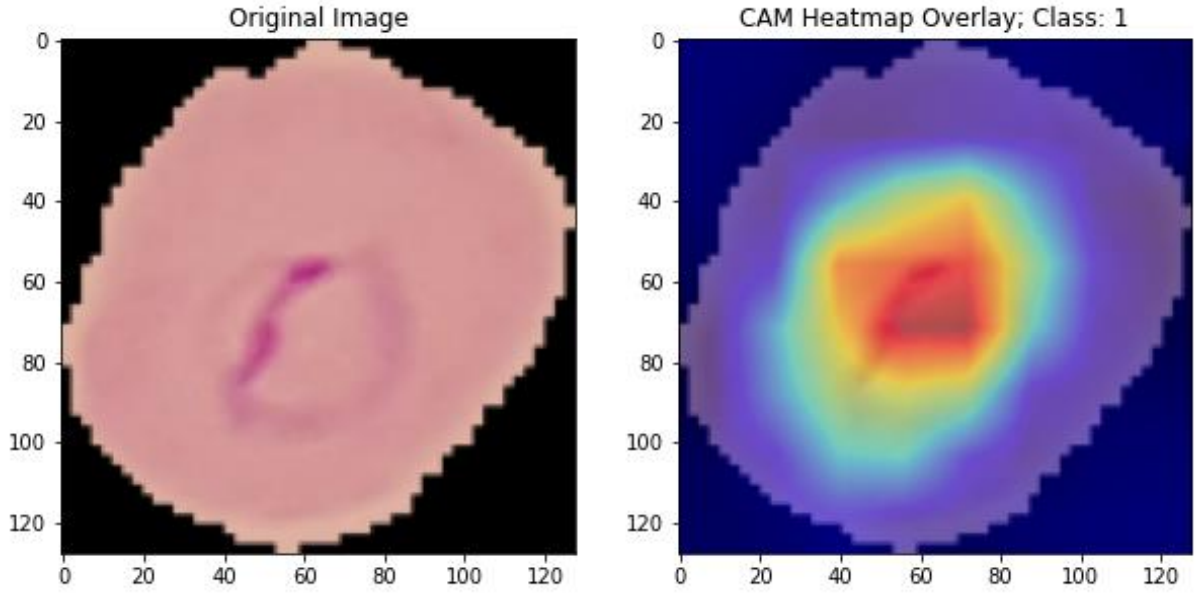
→ Blackbox

Einleitung

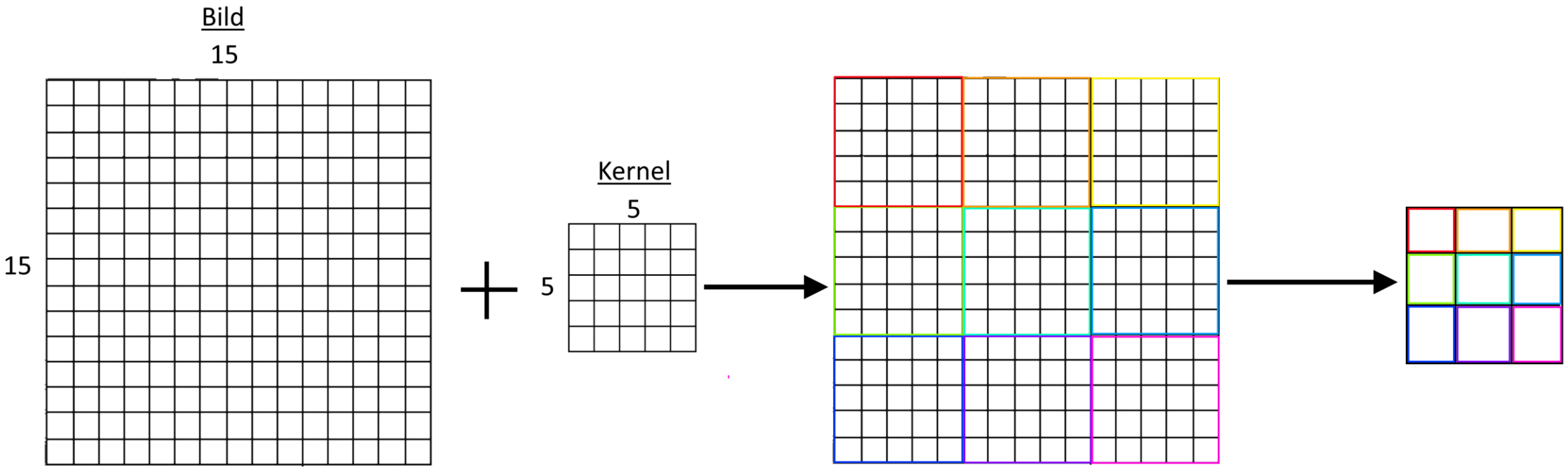
Woran macht ein Neuronales Netz seine Entscheidung fest?

Meistens wissen wir das gar nicht

→ Blackbox

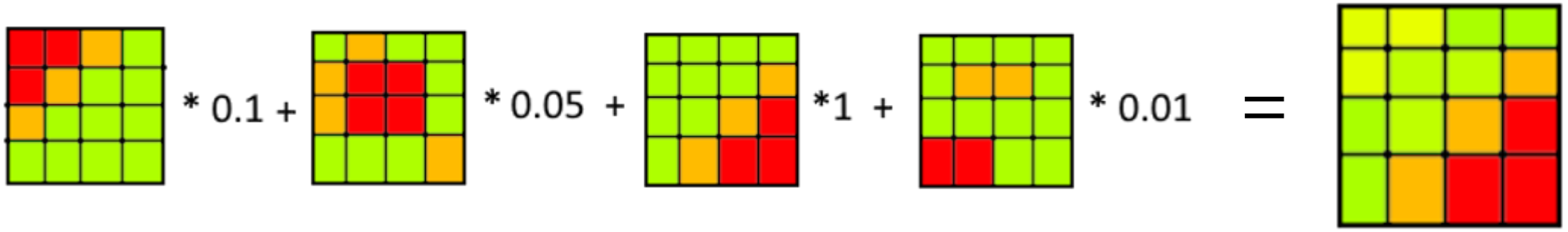


Featuremaps behalten die räumliche Beziehung zum Bild bei!



Class Activation Mapping (CAM)

- Lässt uns die Regionen sichtbar machen, in denen das CNN die Klasse gesehen hat



Implementierung:

```
pred, last_conv_features = model(image, return_cam=True)
```

→ Pred = tensor([[-0.0685, 0.1490]], device='cuda:0', grad_fn=<AddmmBackward0>)

→ last_conv_features = usere ganzen Featuremaps (256 x 14 x 14)

Implementierung:

```
pred, last_conv_features = model(image, return_cam=True)
```

→ Pred = tensor([[-0.0685, 0.1490]], device='cuda:0', grad_fn=<AddmmBackward0>)

→ last_conv_features = usere ganzen Featuremaps (256 x 14 x 14)

```
fc_weights = model.classifier[-1].weight.data
```

→ Gewichte der featuremaps (2 x 256)

Implementierung:

```
pred, last_conv_features = model(image, return_cam=True)
```

→ Pred = tensor([[-0.0685, 0.1490]], device='cuda:0', grad_fn=<AddmmBackward0>)

→ last_conv_features = usere ganzen Featuremaps (256 x 14 x 14)

```
fc_weights = model.classifier[-1].weight.data
```

→ Gewichte der featuremaps (2 x 256)

```
 _, predicted_class = torch.max(pred, 1)
```

→ Vorhergesagte Klasse

Implementierung:

```
pred, last_conv_features = model(image, return_cam=True)
```

→ Pred = tensor([[-0.0685, 0.1490]], device='cuda:0', grad_fn=<AddmmBackward0>)

→ last_conv_features = usere ganzen Featuremaps (256 x 14 x 14)

```
fc_weights = model.classifier[-1].weight.data
```

→ Gewichte der featuremaps (2 x 256)

```
 _, predicted_class = torch.max(pred, 1)
```

→ Vorhergesagte Klasse

```
class_weights = fc_weights[predicted_class]
```

→ Nur die Weights für unsere Klasse

Implementierung:

```
pred, last_conv_features = model(image, return_cam=True)
```

→ Pred = tensor([[-0.0685, 0.1490]], device='cuda:0', grad_fn=<AddmmBackward0>)

→ last_conv_features = usere ganzen Featuremaps (256 x 14 x 14)

```
fc_weights = model.classifier[-1].weight.data
```

→ Gewichte der featuremaps (2 x 256)

```
 _, predicted_class = torch.max(pred, 1)
```

→ Vorhergesagte Klasse

```
class_weights = fc_weights[predicted_class]
```

→ Nur die Weights für unsere Klasse

- ```
  cam = torch.zeros(14,14)
  ```

- ```
  for i, w in enumerate(class_weights[0]):
        cam += w * last_conv_features[0, i]
  ```

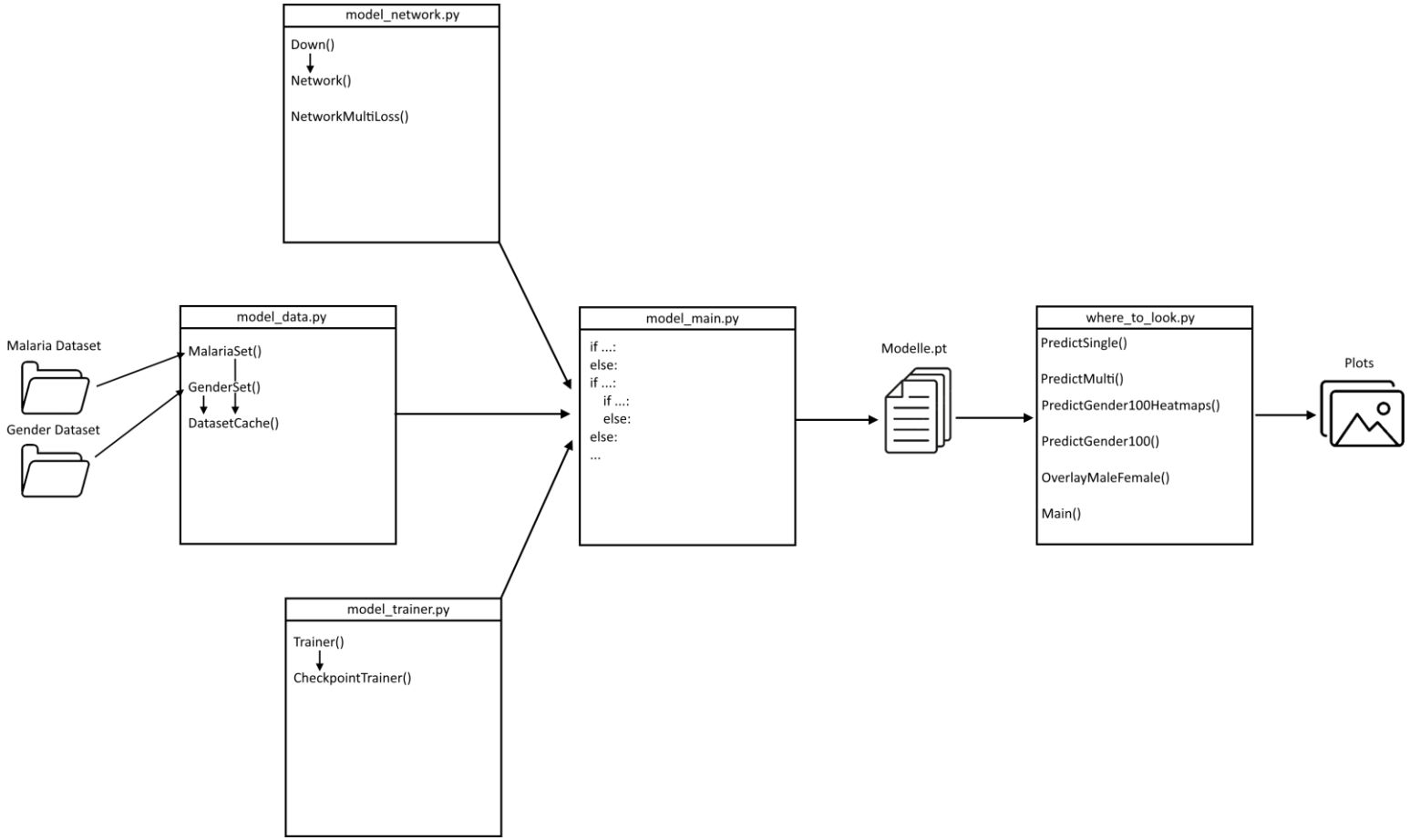
Was kann das Programm / wo liegen Limitierungen?

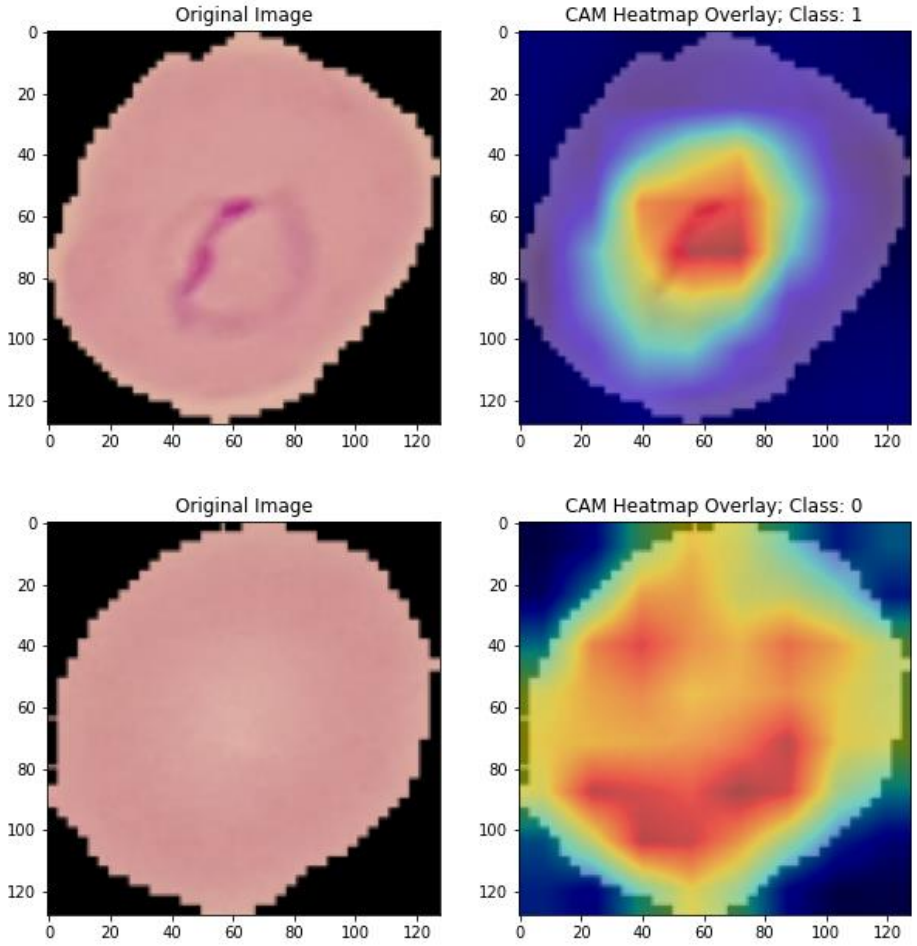
- Was kann es?
  - Eigene Netze trainieren
  - Vollkommen in der Konsole ausführbar
  - Kann Modelle automatisch von Google Drive herunterladen
  - Kann CAM auf verschiedene Weise anwenden
- Limitierungen:
  - Braucht Cuda
  - Kann keine eindeutigen aussagen darüber treffen, was an einem Mann männlich und an einer Frau weiblich ist, da verschiedene gleich trainierte Modelle verschiedene Bereiche anzeigen
  - Binärklassifikation

Was kann das Programm / wo liegen Limitierungen?

- Was kann es?
  - Eigene Netze trainieren
  - Vollkommen in der Konsole ausführbar
  - Kann Modelle automatisch von Google Drive herunterladen
  - Kann CAM auf verschiedene Weise anwenden
- Limitierungen:
  - Braucht Cuda
  - Kann keine eindeutigen aussagen darüber treffen, was an einem Mann männlich und an einer Frau weiblich ist, da verschiedene gleich trainierte Modelle verschiedene Bereiche anzeigen
  - Binärklassifikation

Programm Struktur

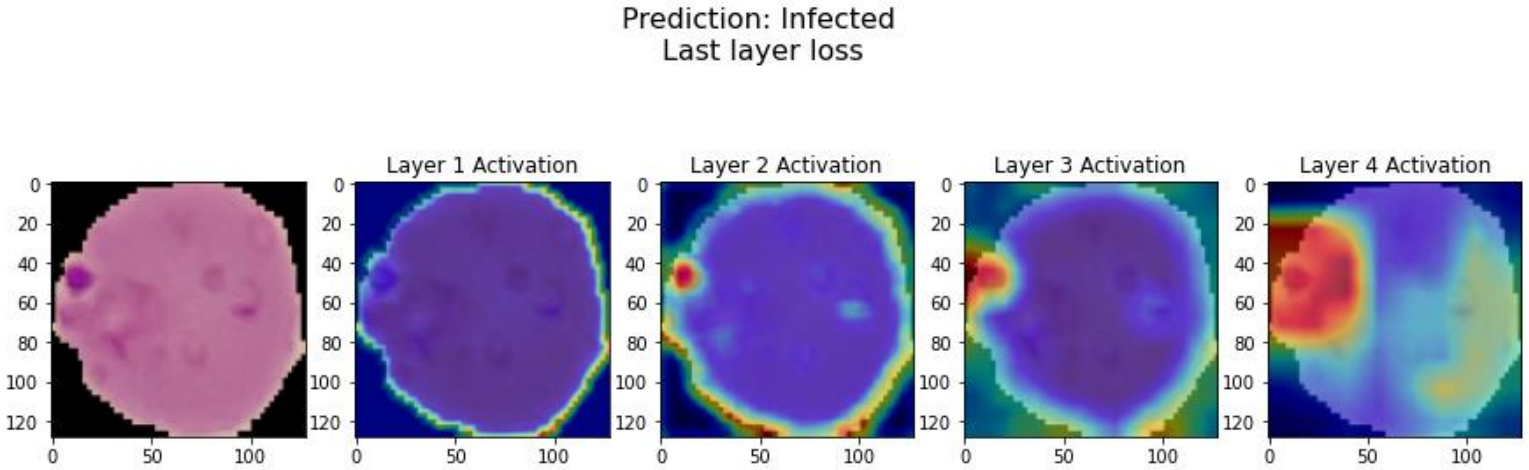


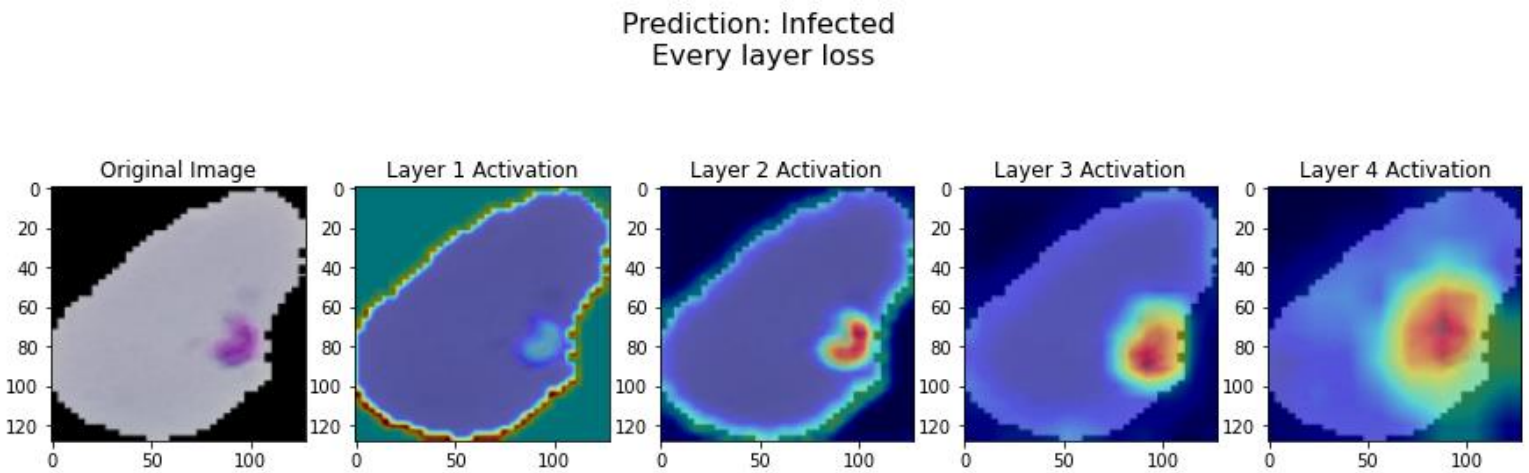


Aktivierungen der letzten Faltungsebene

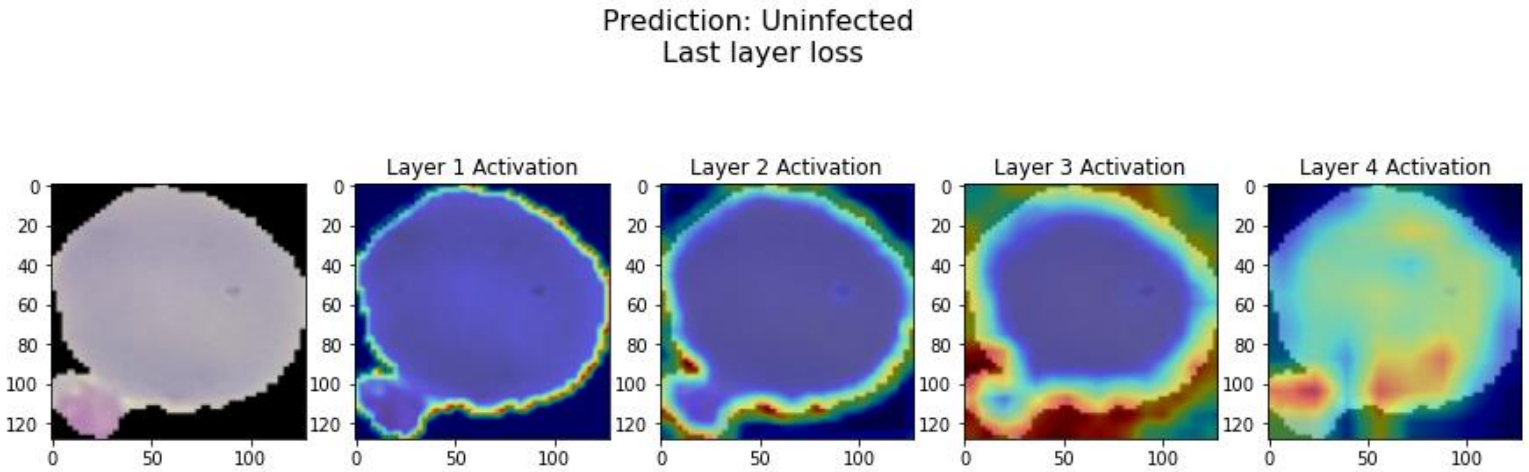
- Gute Lokalisierung bei der infizierten Zelle
- Gesunde Zelle weist kein besonderes Merkmal auf
  → Die Nicht-Existenz eines besonderen Merkmals wird angezeigt

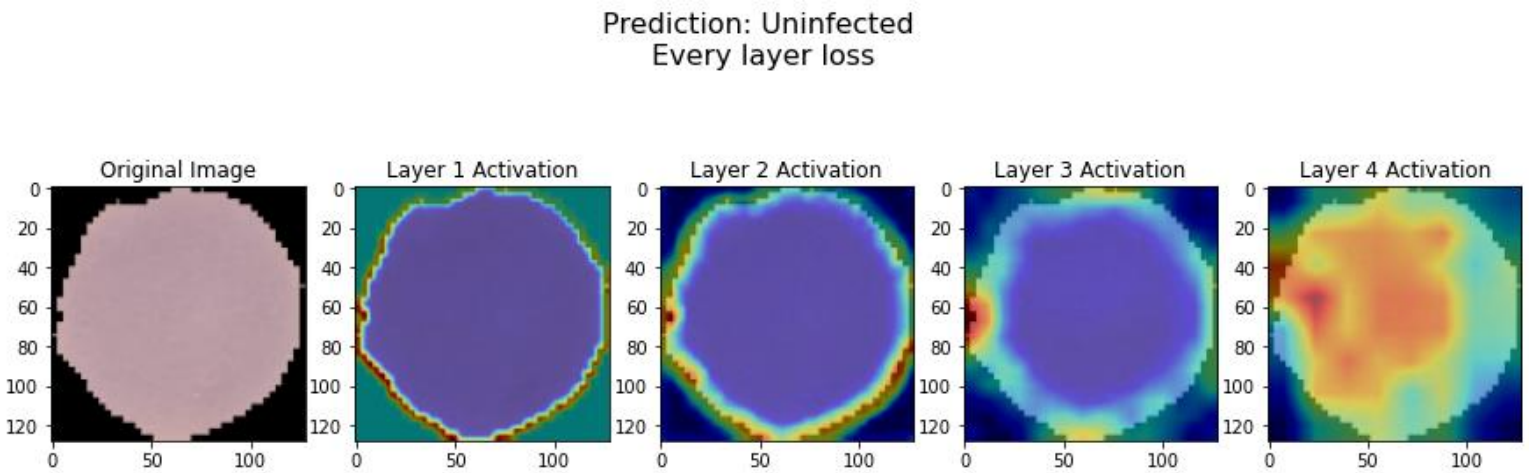
Aktivierungen aller Layer

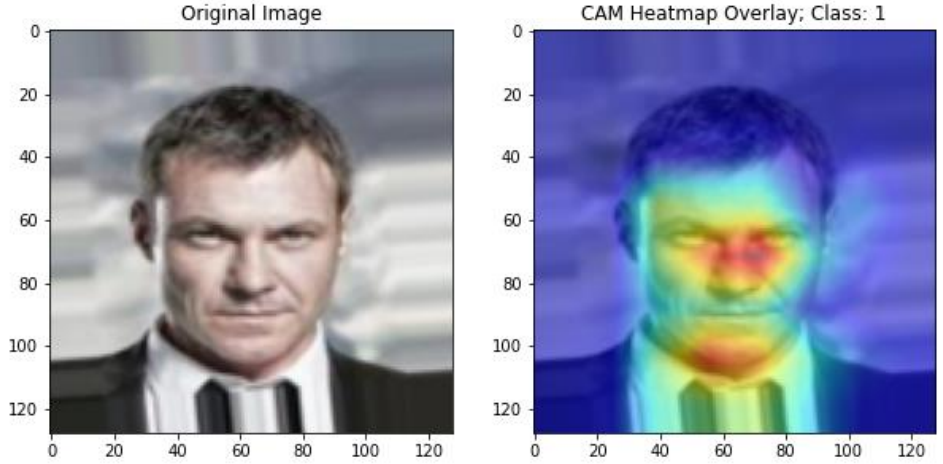


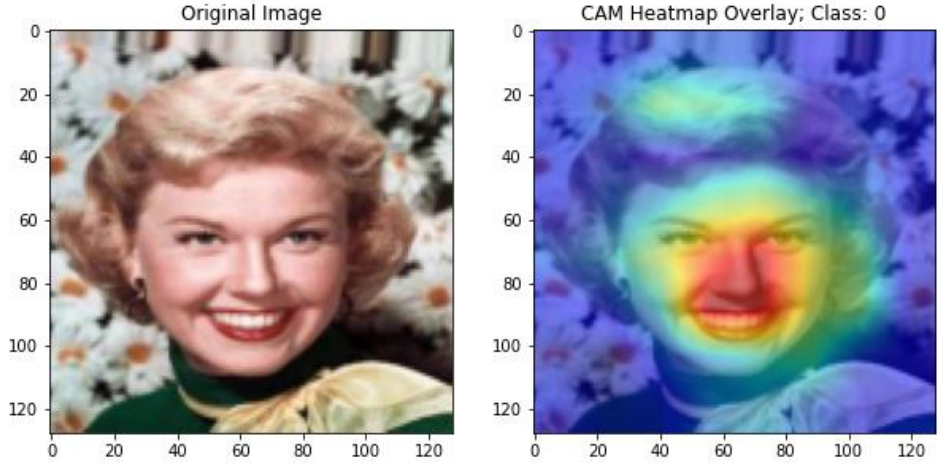


Aktivierungen aller Layer



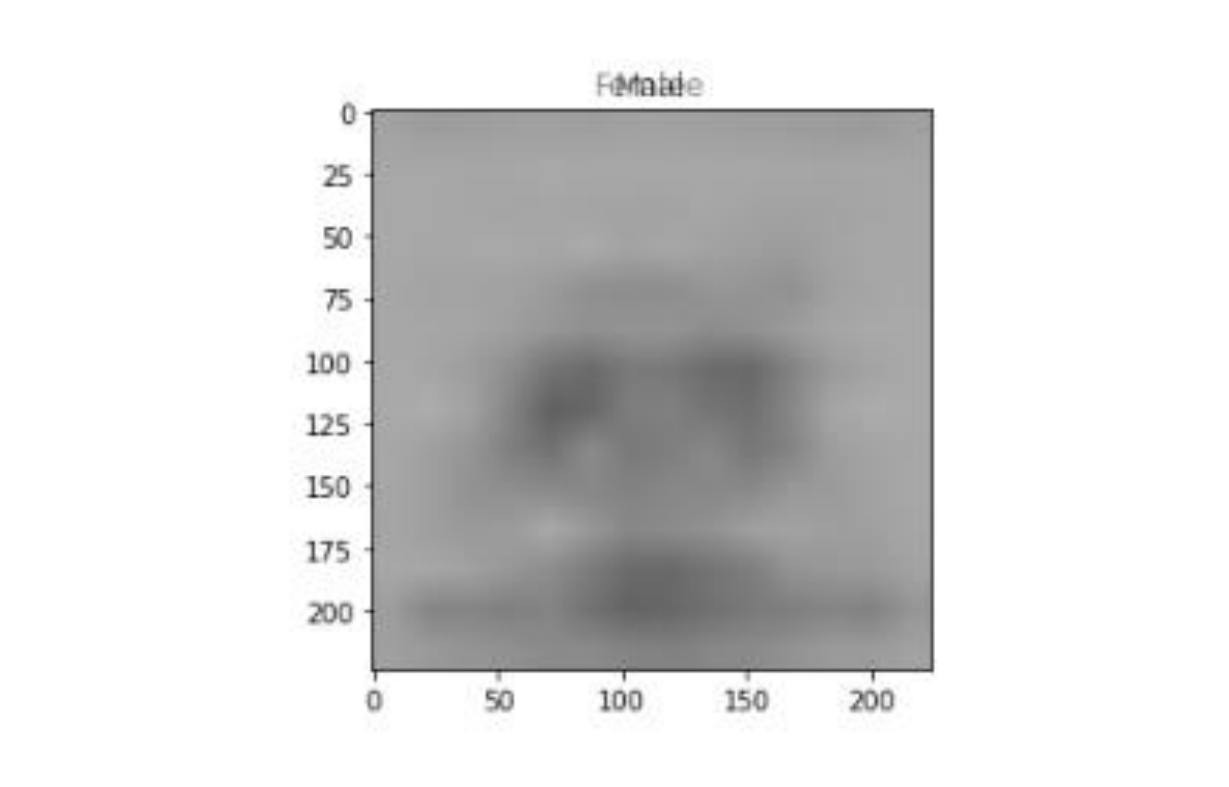


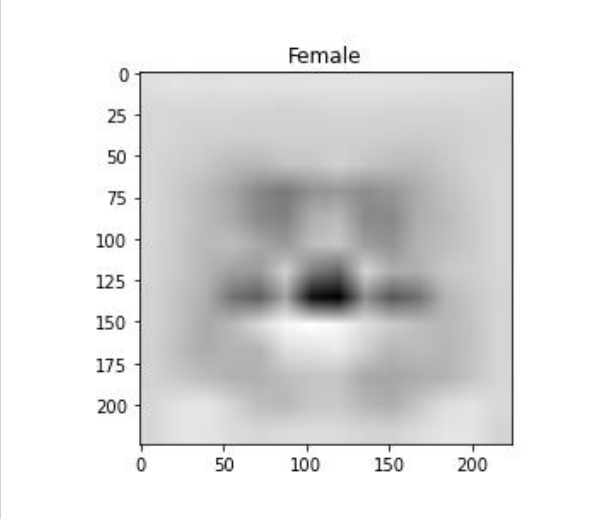


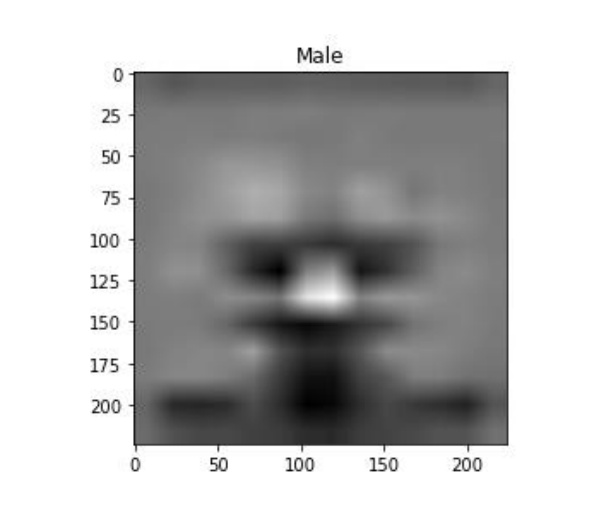


Gender classification

- Der Mann wird eher am oberen Teil der Nase und am Kinn erkannt
- Die Frau an der Mund- Nasen Partie
- Kommt sehr auf das Modell an

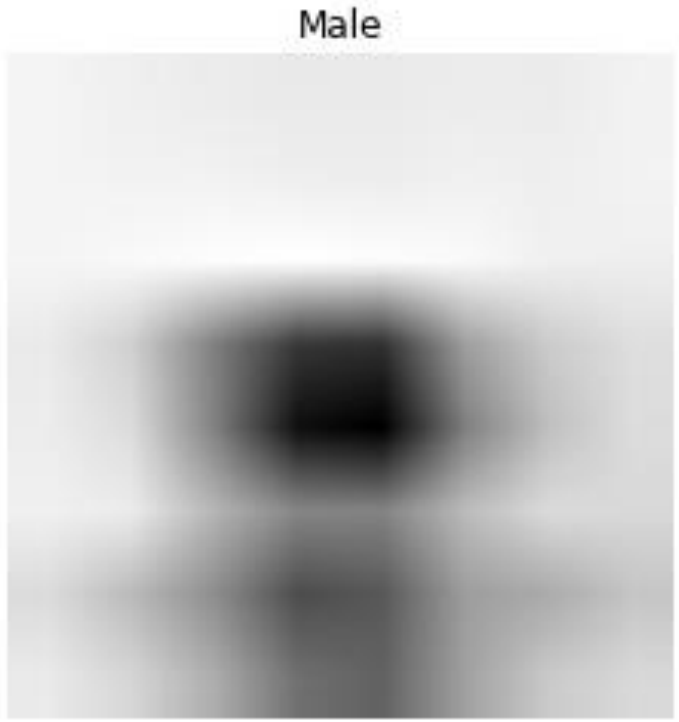


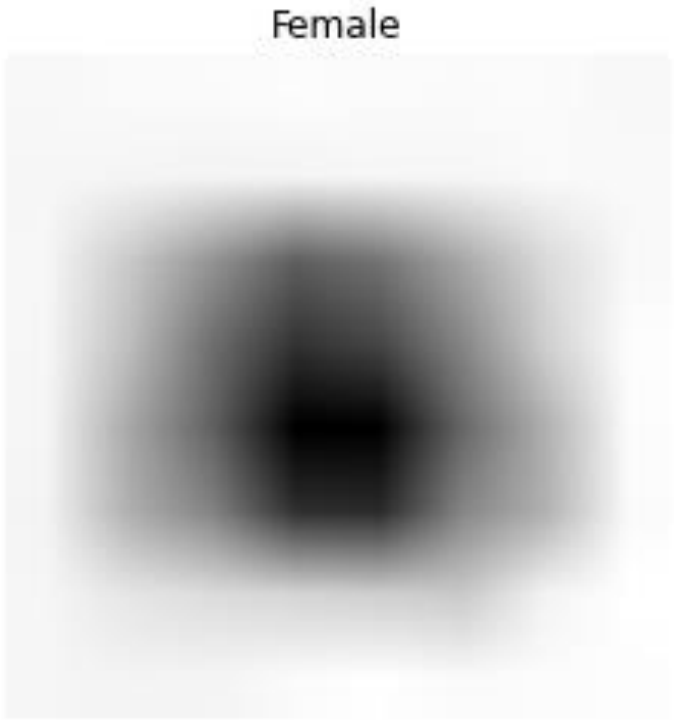




100 Männer/Frauen übereinander geplottet

- Frauen klar an der Nase erkennbar; Teilweise Augenparte & Wangenknochen
- Männer klar am Kinn und den Wangenknochen erkennbar
- Alle Aktivierungen innerhalb einer Silhouette eines Menschen





100 Männer/ Frauen mit einem anderen Modell

- Kaum Unterschiede zu erkennen
- Aktivierungen überlappen sich
- Sehr stark vom Modell abhängig