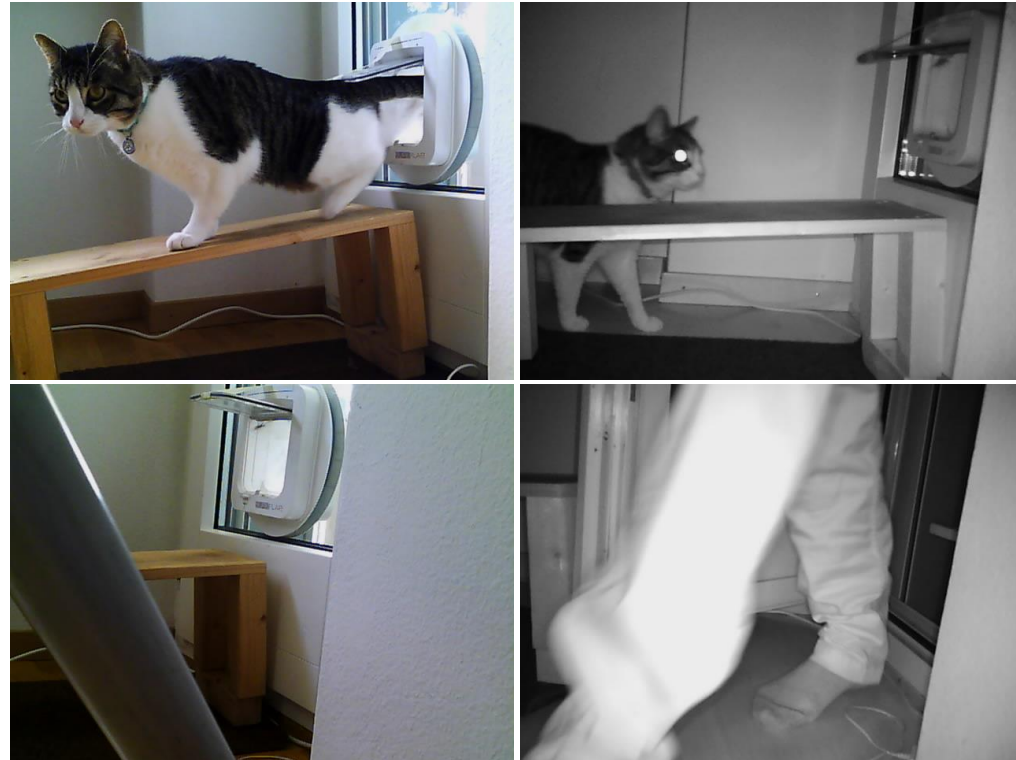


# Cat / No Cat: Bilderkennung anhand von Daten einer Überwachungskamera

- **Team Gruppe 04**
  - Stephan Benke-Bruderer, Michael Hilti, Doris Zahnd
- **Business Case**
  - Michael verwendet zur Überwachung einer Katzentüre eine mit WiFi-verbundene Kamera. Diese reagiert nimmt während Tag und Nacht aufgrund von Bewegungen sechs Bilder auf legt diese ab. Über einen Zeitraum von rund 1.5 Jahren haben sich so rund 85'000 Bilder angesammelt. Aufgrund einer Stichprobe dürften rund 50% keine Katzen, sondern Staubsauger, Füße, Lichteinflüsse etc. enthalten.
  - Ziel ist sämtliche Bilder automatisch in die Kategorien «cat» oder «nocat» sortieren zu lassen.
  - Die Idee ist mittels verschiedener deep learning Netzwerke das Beste für diesen spezifischen Task zu evaluieren.
- **Ablaufschritte**
  - E-Mail und Attachment extraction
  - Evaluation dreier verschiedener Netzwerke: MobileNet, NASNetMobile, Xception
  - Transfer Learning mit Xception
  - Filter
  - Erkenntnisse und abschliessende Bemerkungen
- **Datasets**
  - Nicht öffentliche Bildersammlung mit 85'000 Bildern.

# E-Mail Retrieval, Attachment Extraction, Labelling

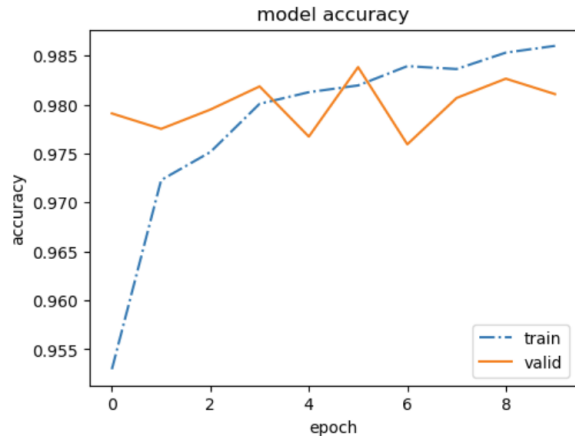
- Script 01 extrahiert 14'491 EML-Dateien aus Gmail.
- Script 02 extrahiert 85'000 Attachments aus den EML-Dateien. Die Bilder sind JPG, 640\*480 Pixel Gross und im RGB Farbraum. Der Datensatz ist über 2.7gb gross.
- Script 04 sortiert Bilder manuell in drei Kategorien.
- (<https://github.com/Nestak2/image-sorter2>)
- Label «cat» = mindestens Teil von Katze sichtbar.
- Nach 3h Aufwand Trainingsdaten- und Validierungsset mit:
  - 4746 «cat»
  - 10922 «nocat»
  - 175 «ausschuss»



# Vergleich der drei Netzwerke

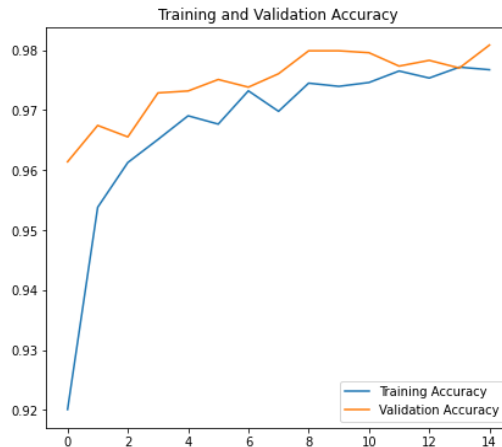
## Xception

- 22.9 m Parameters, top-5 accuracy bei 0.945, 88 MBytes gross
- Xception + FC + Dropout + FC



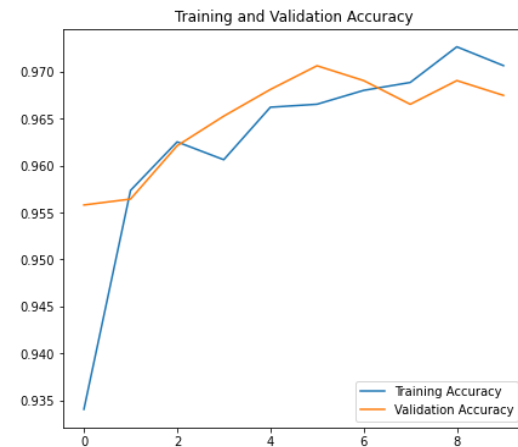
## MobileNet

- 4.2 m Parameters, top-5 accuracy bei 0.895, 16 MBytes gross
- MobileNet + FC + Dropout + FC (Classification)



## NASNetMobile

- 5.3 m Parameters, top-5 accuracy bei 0.919, 23 MBytes gross
- Auf Datensatz mit 20 Epochen 0.98 accuracy, ohne drop-out und FC.



# Falsch-Klassifikationen

(Sample Bilder vom MobileNet Validation Set)

20180117\_125257\_05.jpg  
WRONG  
ground truth: NOCAT  
predicted: CAT  
probability=0.82



20180117\_135336\_04.jpg  
WRONG  
ground truth: NOCAT  
predicted: CAT  
probability=0.85



20180107\_110139\_05.jpg  
WRONG  
ground truth: NOCAT  
predicted: CAT  
probability=0.88



20180106\_190956\_04.jpg  
WRONG  
ground truth: CAT  
predicted: NOCAT  
probability=0.5



20180120\_080330\_03.jpg  
WRONG  
ground truth: CAT  
predicted: NOCAT  
probability=0.52



20180117\_192751\_01.jpg  
WRONG  
ground truth: CAT  
predicted: NOCAT  
probability=0.52



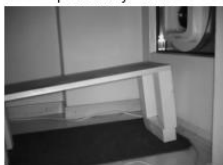
20180107\_130646\_04.jpg  
WRONG  
ground truth: NOCAT  
predicted: CAT  
probability=0.91



20180117\_064306\_04.jpg  
WRONG  
ground truth: NOCAT  
predicted: CAT  
probability=0.92



20180105\_114058\_01.jpg  
WRONG  
ground truth: NOCAT  
predicted: CAT  
probability=0.92



20180103\_175334\_04.jpg  
WRONG  
ground truth: CAT  
predicted: NOCAT  
probability=0.63



20180102\_101544\_03.jpg  
WRONG  
ground truth: CAT  
predicted: NOCAT  
probability=0.64



20180108\_201843\_02.jpg  
WRONG  
ground truth: CAT  
predicted: NOCAT  
probability=0.71



20180120\_150703\_02.jpg  
WRONG  
ground truth: NOCAT  
predicted: CAT  
probability=1.0



20180102\_114532\_05.jpg  
WRONG  
ground truth: NOCAT  
predicted: CAT  
probability=1.0



20180120\_134650\_01.jpg  
WRONG  
ground truth: NOCAT  
predicted: CAT  
probability=1.0



20180105\_005725\_03.jpg  
WRONG  
ground truth: CAT  
predicted: NOCAT  
probability=0.75



20180101\_143857\_06.jpg  
WRONG  
ground truth: CAT  
predicted: NOCAT  
probability=0.77



20180101\_124910\_02.jpg  
WRONG  
ground truth: CAT  
predicted: NOCAT  
probability=0.79



# Layer von VGG16 visualisieren

Use gradient ascent to visualize images that maximally activate specific filters from different layers of the model.

<https://www.coursera.org/projects/visualizing-filters-cnn-tensorflow>

<https://blog.keras.io/how-convolutional-neural-networks-see-the-world.html>

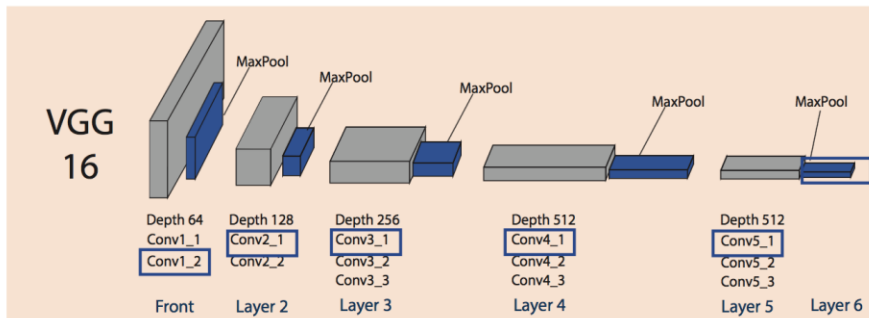
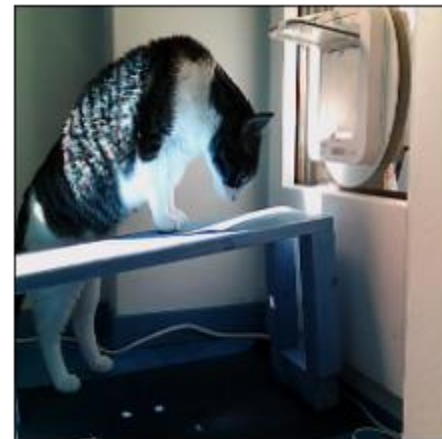
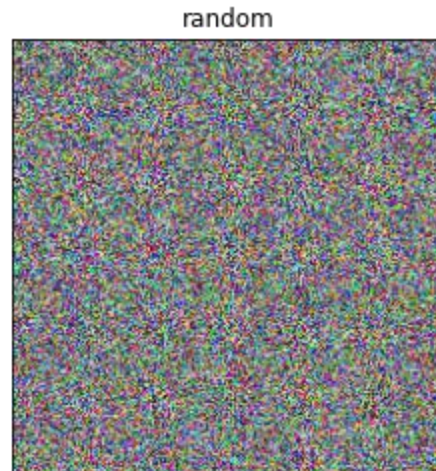


Image credit:

<https://medium.datadriveninvestor.com/creating-art-through-a-convolutional-neural-network-ed8a4d9a3f87>

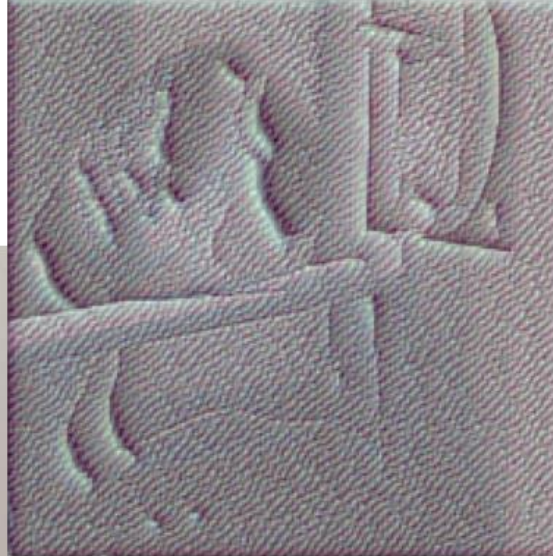




# Layer von VGG16 visualisieren

block2\_conv1, filter index 90

Custom Image



Random Image



block1\_conv1, filter index 39

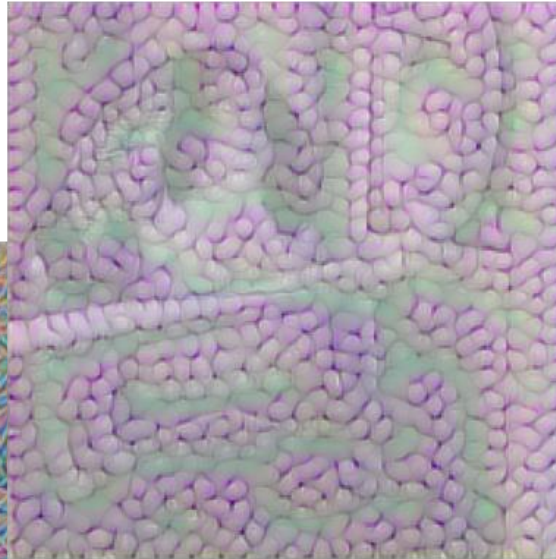
Custom Image



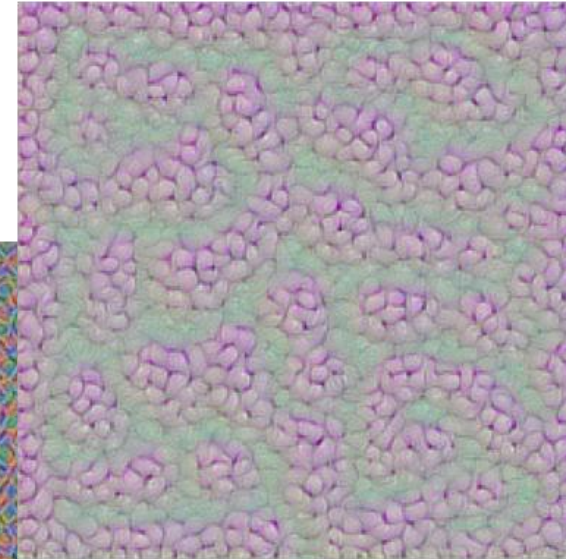
# Layer von VGG16 visualisieren

block3\_conv2, filter index 102

Custom Image

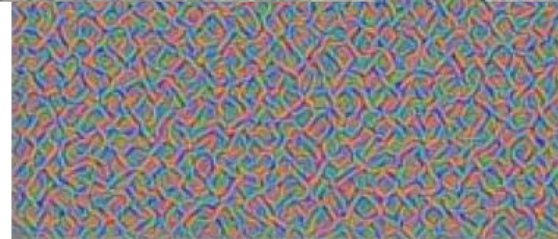
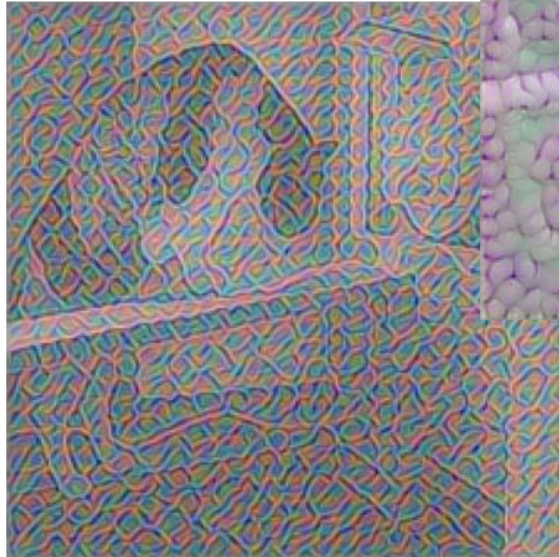


Random Image



block3\_conv1, filter index 134

Custom Image





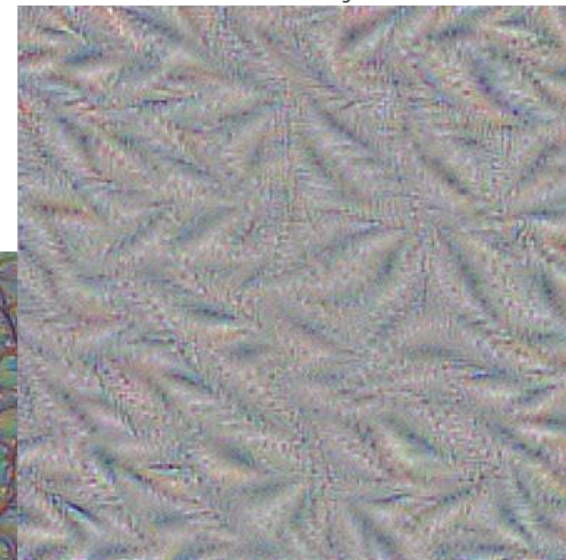
# Layer von VGG16 visualisieren

block4\_conv3, filter index 221

Custom Image

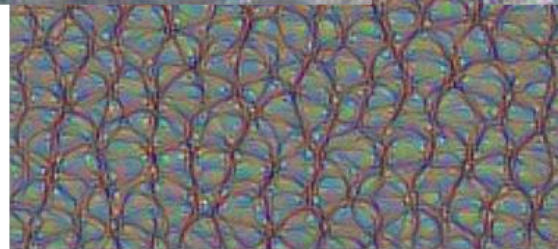
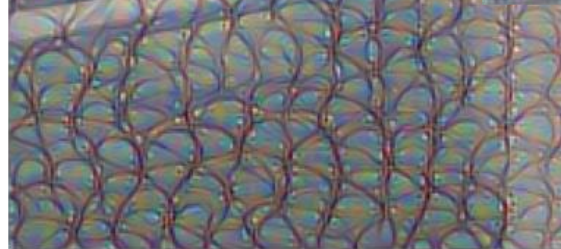
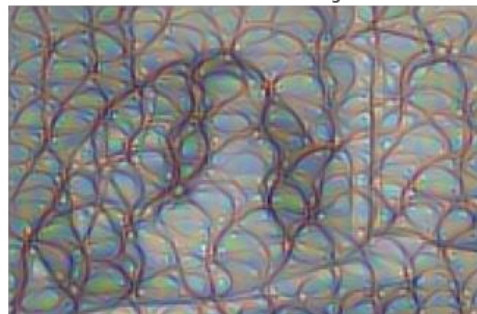


Random Image



block4\_conv1, filter index 230

Custom Image





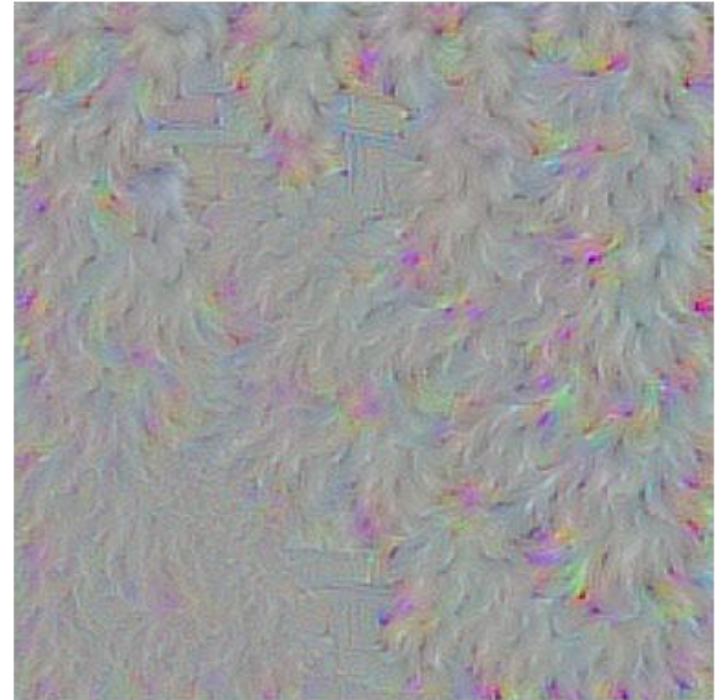
# Layer von VGG16 visualisieren

block5\_conv1, filter index 339

Custom Image



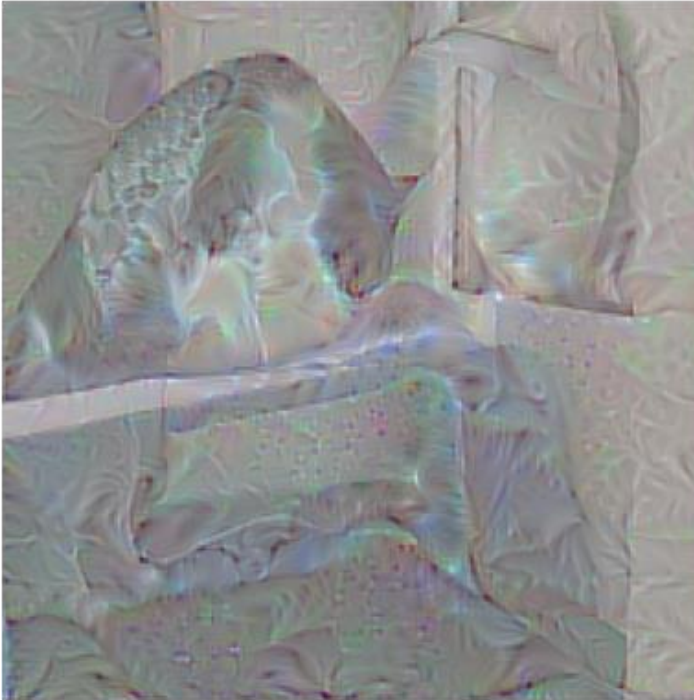
Random Image



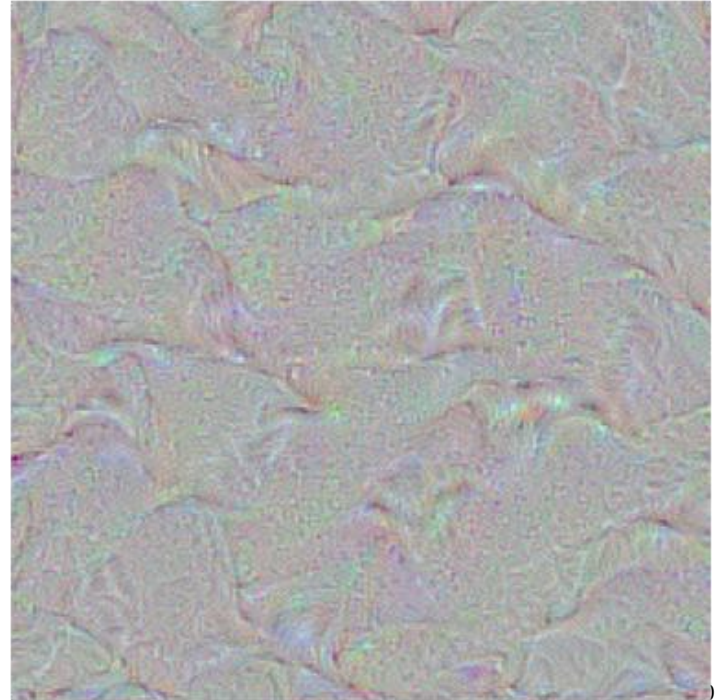
# Layer von VGG16 visualisieren

block5\_conv3, filter index 113

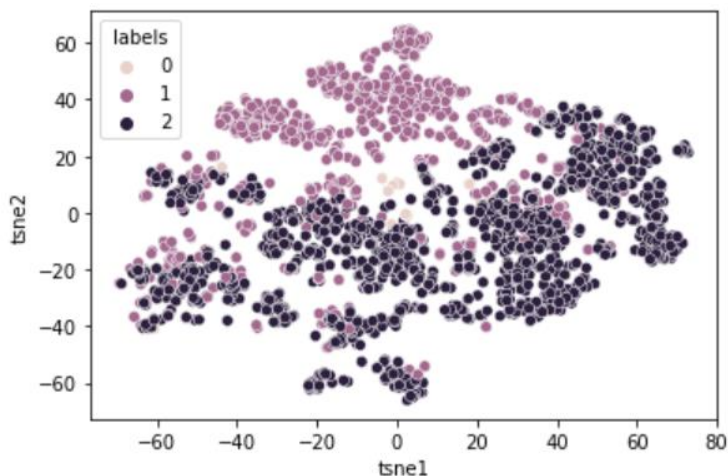
Custom Image



Random Image



# Transfer Learning (Xception)

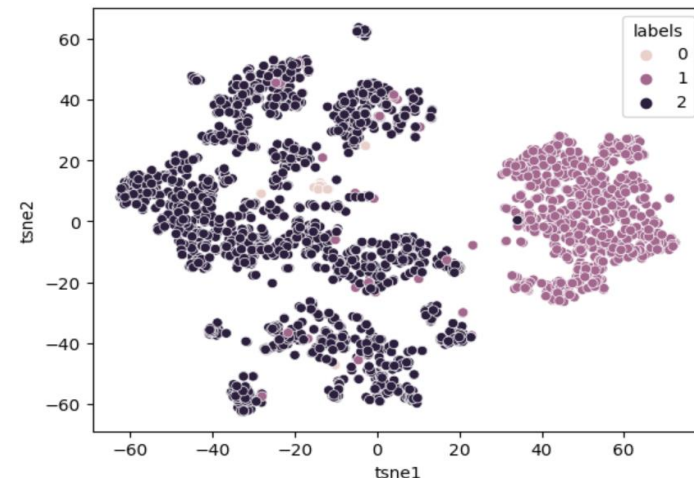


Features from last pooling layer.  
Pretrained Xception net.

+ 1 fully connected  
layer (w/ dropout)



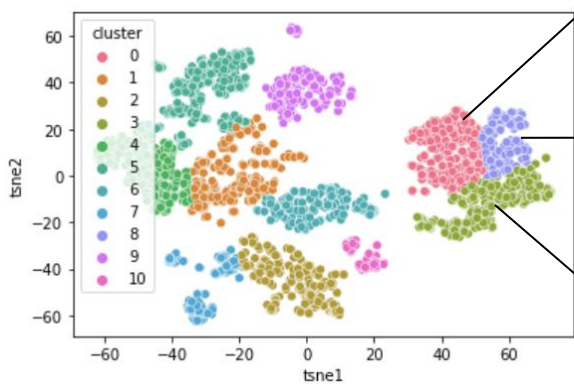
Layer trained  
for 10 epochs



Features from last fully  
connected layer.



# Gradient CAMs: Cats

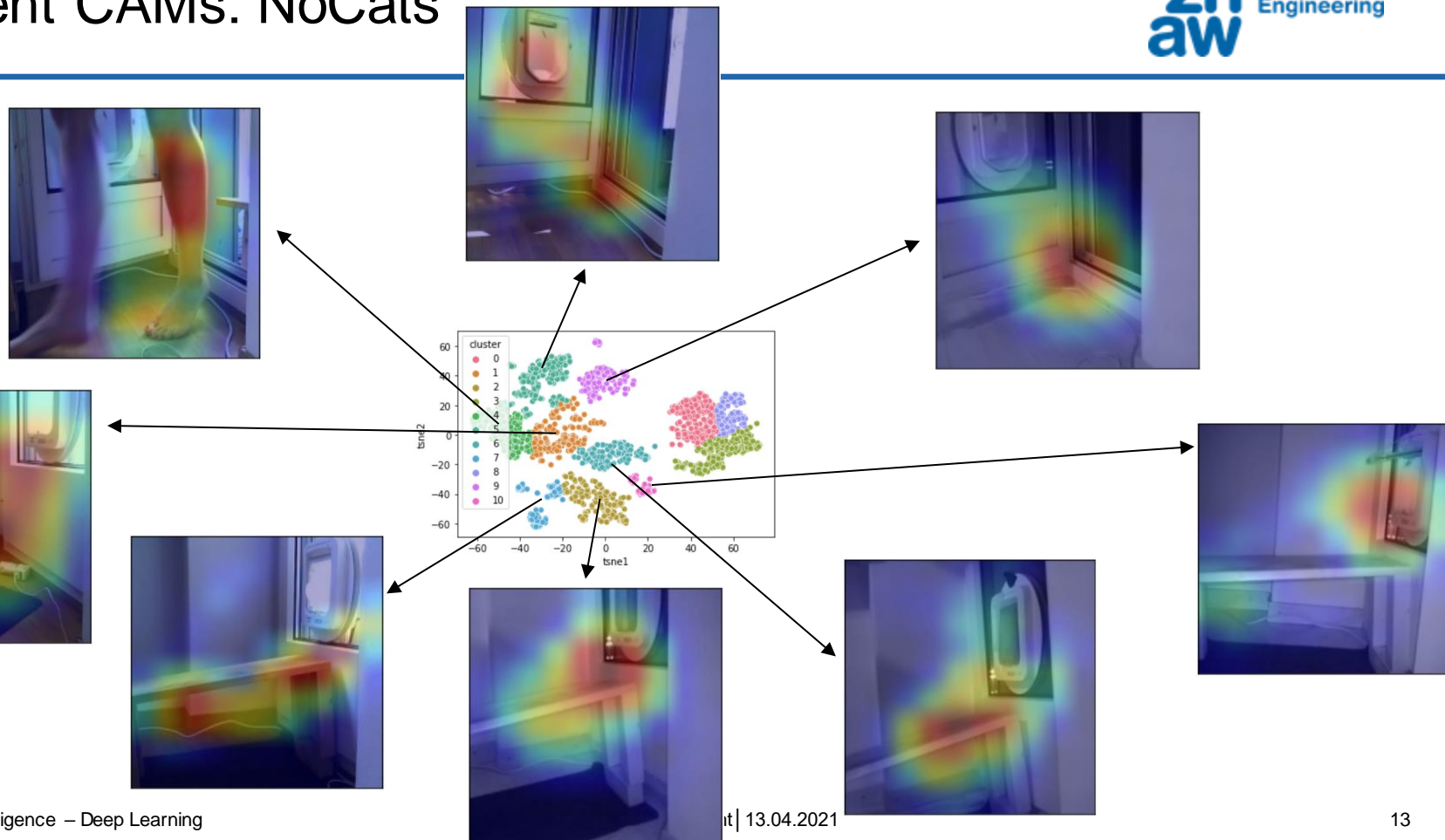


Free labels from features?



Net seems to be triggered by cat head and legs.

# Gradient CAMs: NoCats



# Erkenntnisse und abschliessende Betrachtungen

---

- Grösser nicht unbedingt besser: auf dem verwendeten Datensatz erreichten alle drei verwendeten Modelle sehr gute Resultate. Auch sehr kleine Modelle und daher schnell laufende Modelle erreichen brauchbare Resultate.
- Colab einfach und praktisch, aber nicht immer gleich schnell und stabil.
- Aufwand für Labeling in diesem Fall sehr tief, weil keine Bildbereiche gelabelt werden mussten. Hängt stark vom Datensatz ab, in diesem Bild gut funktioniert.
- Ganz grundsätzlich: beeindruckend was out-of-the-box möglich ist.



# Weiterführende Links

---

- Email Extractor von Github User Andrew Jennings: <https://github.com/abjennings/gmail-backup>
- Attachmenet Extractor von Github User von Stephan Hügel: <https://gist.github.com/urschrei/5258588>
- Image Sorter von Github User Nestor Arsenov: <https://github.com/Nestak2/image-sorter2>
- Github Repository des Projekts Cat/No Cat: <https://github.com/gruppeneun/catnocat>