

---



# InceptionNet

Seminar Deep Learning

Jens Ullrich

Informatik Master

Hochschule Offenburg

# Gliederung

- Einleitung: Was ist InceptionNet?
- Besonderheiten:
  - Inception Module
  - Auxiliary Classifier
- Weiterentwicklung: Inception v3
  - Verbesserungen aus Inception v2
  - Zusätzliches in Inception v3
- Experimente

# Was ist InceptionNet?

- Inception:
  - Deep Convolutional Neural Network Architecture
  - Ziel: Optimale Nutzung der Rechenleistung im neuronalen Netz
  - Lösung: Architekturdiseign was es erlaubt, die Tiefe und Breite des Netzes bei konstantem Rechenbudget zu erhöhen
- GoogLeNet (Inception v1):
  - Konkrete Umsetzung der Inception Architektur
  - State of the Art in ImageNet Large-Scale Visual Recognition Challenge 2014
  - 22 Schichten tief (27 wenn man pooling layers mitzählt)
  - Insgesamt ca. 100 Layer

# Besonderheiten

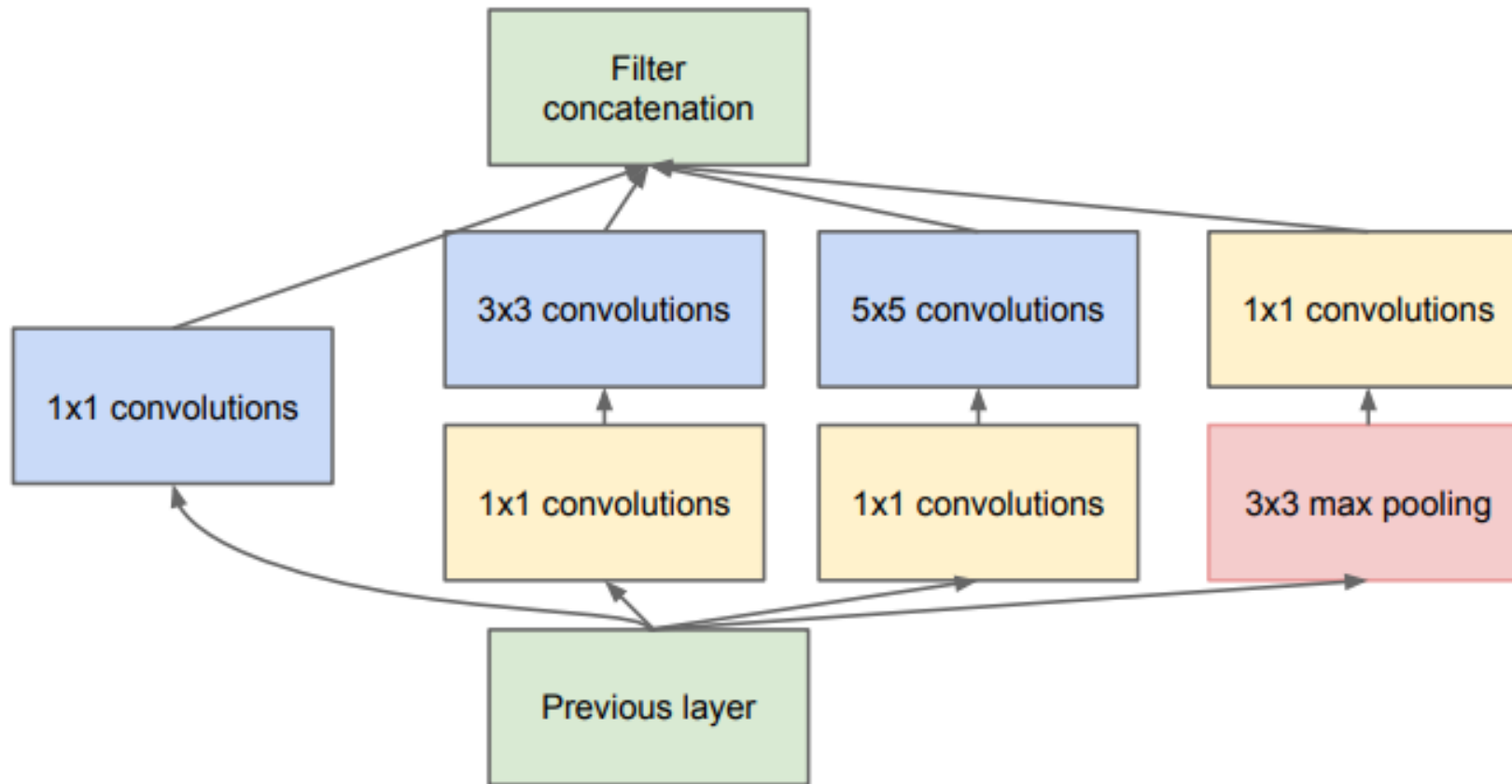
## Probleme, die gelöst werden sollten:

- Wesentliche Bestandteile des Bildes können unterschiedlich groß im Bild sein
- Probleme von sehr tiefen Netzen: overfitting, vanishing gradient
- Steigender Rechenaufwand

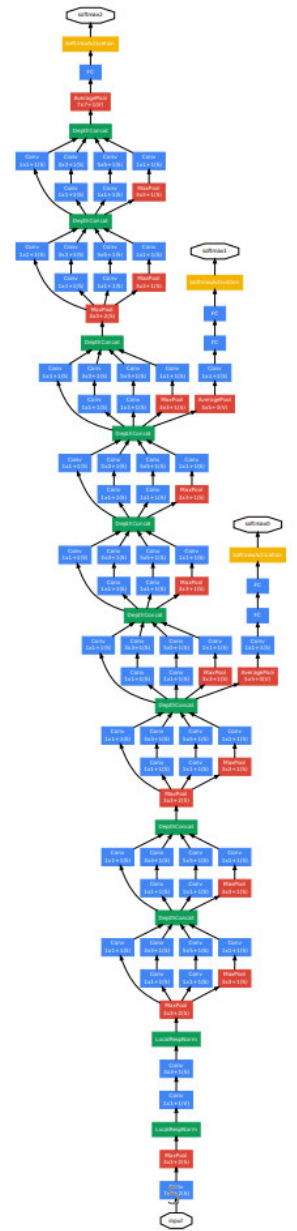
## Lösungsansätze:

- Mehrere Filter verschiedener Größe auf einer Ebene
- Sparsely Connected vs Fully Connected
- Zusätzliche Filter zum Reduzieren der Dimensionen
- Zusammenfassen der Outputs aus den Filtern

# Besonderheiten: Inception Module



Seminar Deep Learning - InceptionNet - Jens Ullrich -  
Informatik Master - Hochschule Offenburg



# Besonderheiten: Auxiliary Classifiers

## Problem:

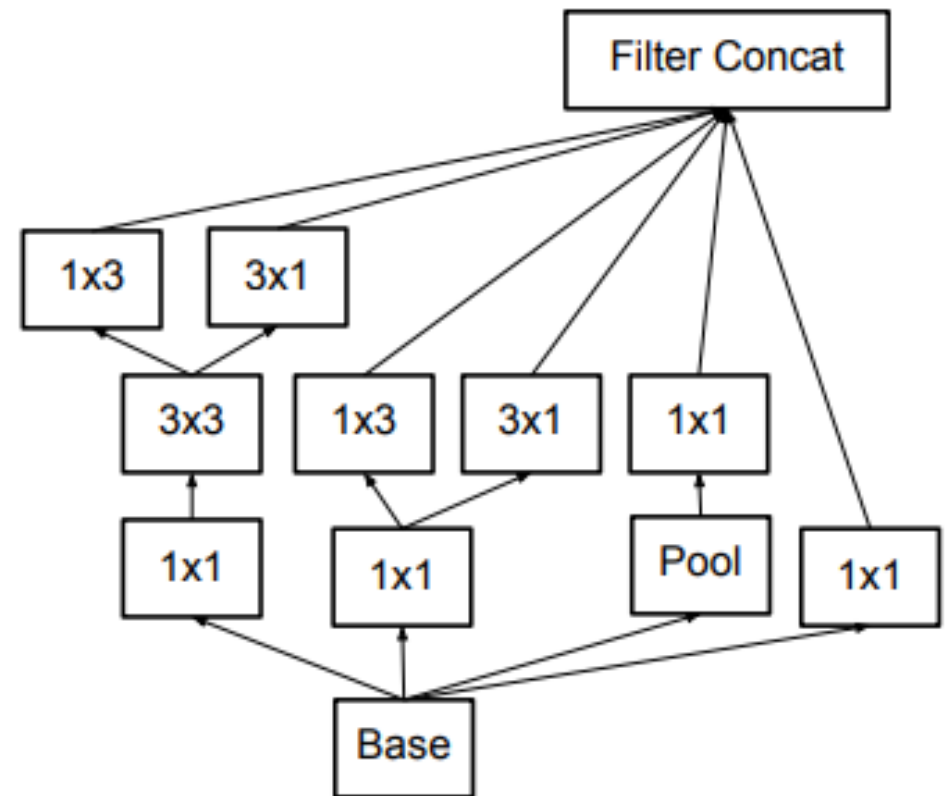
- Gradient Backpropagation durch relative tiefe Netze kann ineffizient werden.

## Lösung:

- Kleine CNNs, die an die mittleren Inception Module angehängt werden
- Total Loss wird aus allen Outputs berechnet
- Loss der Auxiliary Classifier wird dabei mit 0.3 gewichtet
- Auxiliary Classifier werden natürlich **nur beim Training** genutzt

# Weiterentwicklung: Inception v3

- Verbesserungen aus Inception v2:
  - Faktorisierung der größeren Filter
  - Beispiel:  $1 \times 3 + 3 \times 1$ 
    - 33% günstiger als  $3 \times 3$





# Weiterentwicklung: Inception v3

- Zusätzliche Verbesserungen in Inception v3:
  - RMSProp Optimizer
  - BatchNorm in Auxiliary Classifiers
  - Label Smoothing
- Weitere Versionen: Inception v4, Inception-ResNet



# Experimente:

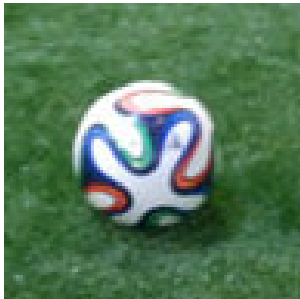
- Code:
  - Pytorch
  - Torchvision Models: GoogLeNet (Inception v1) + Inception (Inception v3)
- Features:
  - Daten im Format *ImageFolder* laden und vorverarbeiten
  - Modelle für InceptionNet v1 und v3 initialisieren
  - Modelle trainieren, Trainings- und Validierungsergebnisse ausgeben
  - Visualisierung mit TensorBoard
  - Confusion Matrix erstellen und visualisieren

# Daten: MagmaDataSet

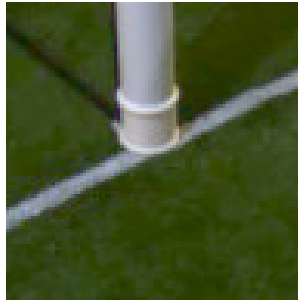
- 8 Klassen
  - Trainset: 4917 Bilder
  - Testset: 1230 Bilder
  - Format: ImageFolder
    - Bilder sind nach Klasse sortiert in Unterordnern gespeichert
- Beispiel:
    - root\_path/train/0/123.png
    - root\_path/train/6/456.png
    - root\_path/train/1/789.png
    - root\_path/train/5/101.png
    - root\_path/test/4/112.png
    - root\_path/test/6/131.png
    - root\_path/test/6/415.png
    - root\_path/test/7/161.png

# Daten: MagmaDataSet

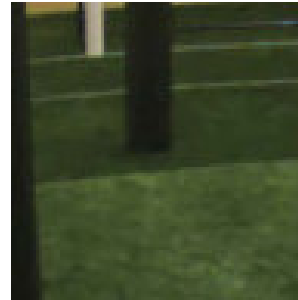
0



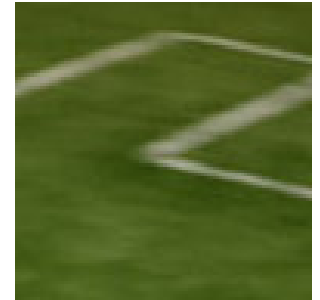
1



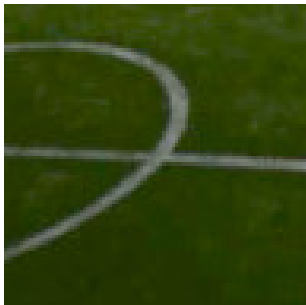
2



3



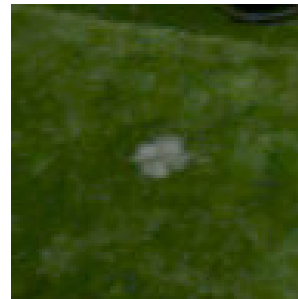
4



5



6



7

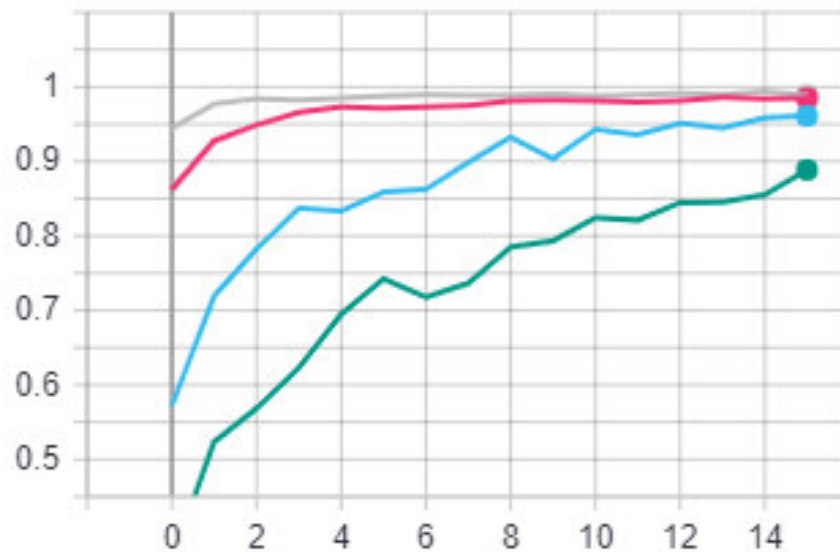


# Experimente: Parameter

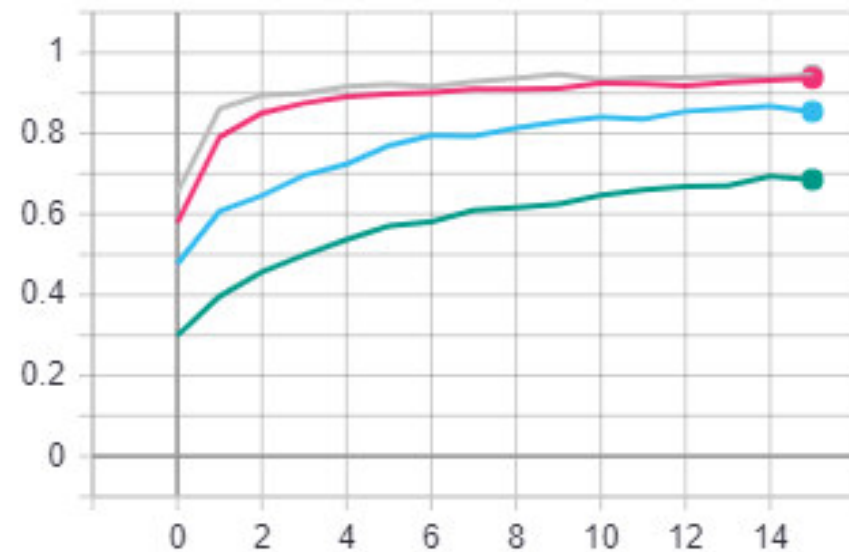
- Folgende Parameter wurden untersucht:
  - Batch Size
  - Learning Rate + Reduction
  - Anzahl Epochen
- Folgende Werte wurden für die folgenden Ergebnisse genutzt:
  - Batch Size: 16
  - Learning Rate: 0.001
  - Anzahl Epochen: 16
- Bestes Ergebnis: InceptionNet v3, vortrainiert (mit ImageNet Daten)
  - Best validation Accuracy: 0.994309

# Ergebnisse: Accuracy

test  
tag: Accuracy/test



train  
tag: Accuracy/train

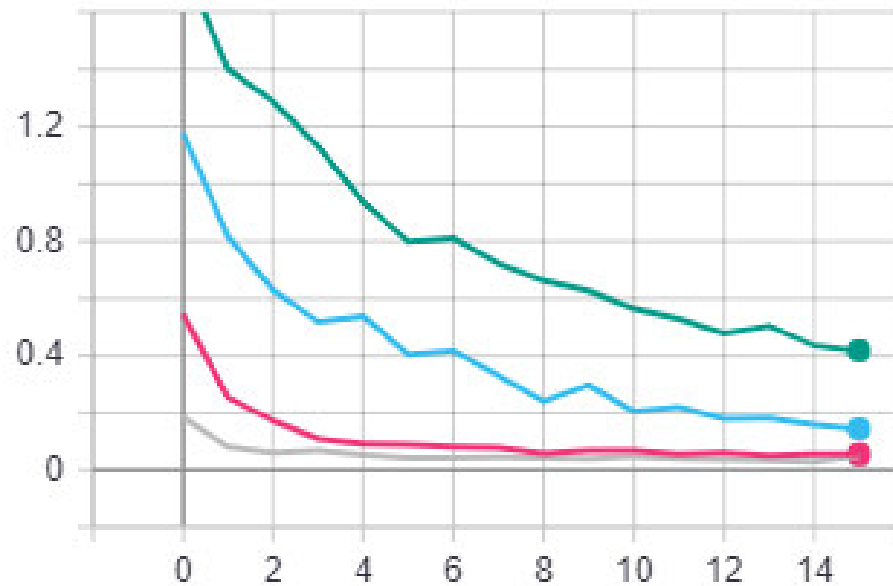


v1 not pretrained / v1 pretrained / v3 not pretrained / v3 pretrained

# Ergebnisse: Loss

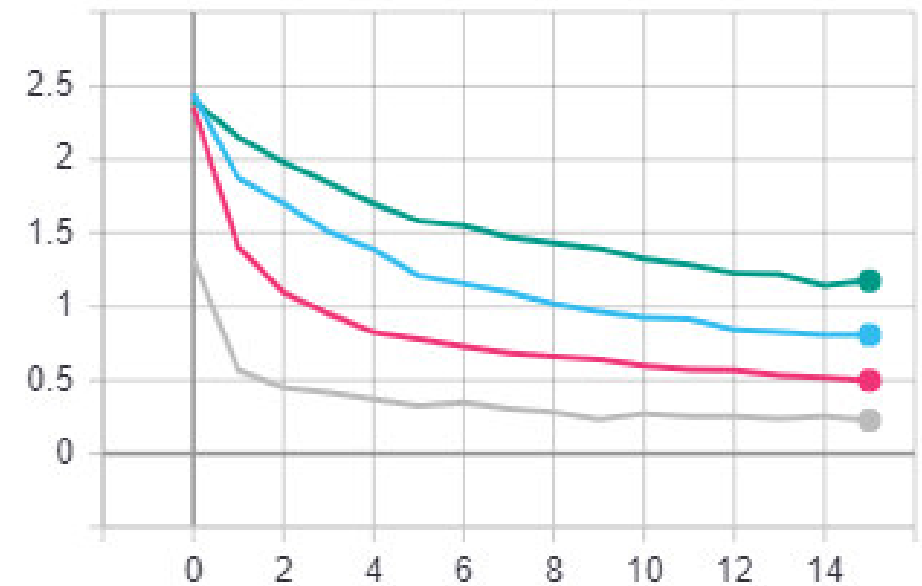
test

tag: Loss/test



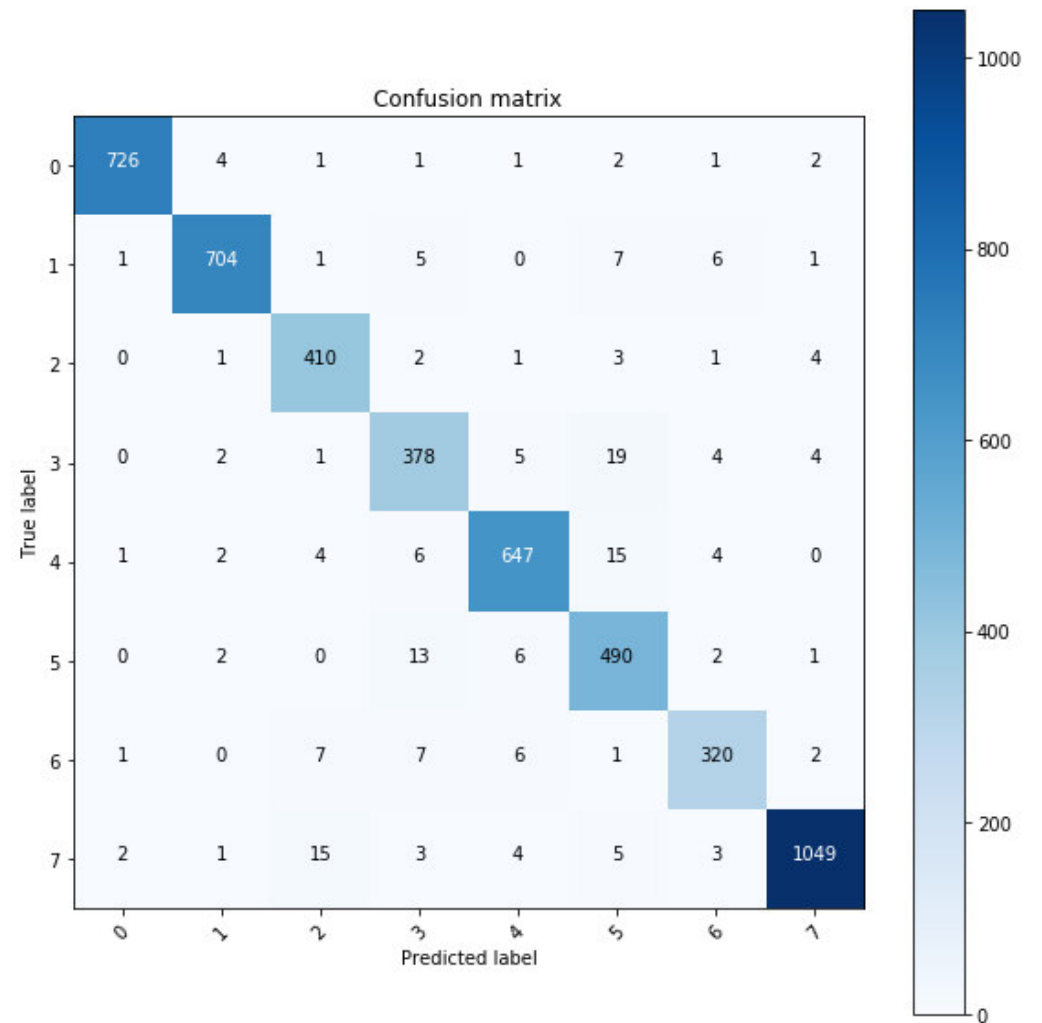
train

tag: Loss/train



v1 not pretrained / v1 pretrained / v3 not pretrained / v3 pretrained

# Ergebnisse: Confusion Matrix







Vielen Dank für  
die  
Aufmerksamkeit!

Links:

- Paper:
  - InceptionNet v1: <https://arxiv.org/pdf/1409.4842v1.pdf>
  - InceptionNet v2 & v3: <https://arxiv.org/pdf/1512.00567.pdf>
- Hilfreiche Artikel:
  - <https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202>
  - <https://www.analyticsvidhya.com/blog/2018/10/understanding-inception-network-from-scratch/>
- Code:
  - <https://github.com/jensullrich/DL-Seminar>
- Tutorials:
  - Using TorchVision Models: [https://pytorch.org/tutorials/beginner/finetuning\\_torchvision\\_models\\_tutorial.html](https://pytorch.org/tutorials/beginner/finetuning_torchvision_models_tutorial.html)
  - TensorBoard: <https://pytorch.org/docs/stable/tensorboard.html>
  - Confusion Matrix: <https://deeplizard.com/learn/video/0LhiS6yu2qQ>