

TECHNICAL APPLICATIONS AND DATA MANAGEMENT. SS 2023.

VORLESUNG 9

15.05.2023
MÜNCHEN
STUDIENGANG
SUSTAINABILITY
MANAGEMENT.



AGENDA

1. Wahl Projektarbeit AI
2. Case Study

GEPLANTE ROADMAP DER VORLESUNG.

ROADMAP	WAS HABEN WIR VOR?
Vorlesung 1	Übersicht und Einführung
Vorlesung 2	Einführung Data Science und Data Science Workflow, Detaillierung Data Engineering
Vorlesung 3 und Vorlesung 4	Deskriptive und explorative Datenanalyse und Vertiefung anhand Case Study
	Vertiefung Datenanalyse anhand Case Study
Vorlesung 5	Aufgabenstellung Data Science, Übersicht und Einführung Machine Learning, unüberwachtes Lernen
Vorlesung 6 und Vorlesung 7	Überwachtes Lernen
	Vertiefung überwachtes Lernen anhand Case Study
Vorlesung 8 und Vorlesung 9	Neuronale Netze und Convolutional Neural Networks (CNN)
	Vertiefung CNN anhand Case Study, Aufgabenstellung AI
Vorlesung 10	Rekurrente Neuronale Netze
Vorlesung 11	Generative AI
Vorlesung 12	Ausblick
Vorlesung 13	„Fragestunde“

Folien der bisherigen Vorlesung verfügbar unter [Link](#)

1. PROJEKTARBEIT AI

WAHL PROJEKTARBEIT: FOLGENDE THEMEN STEHEN FÜR DEN AI-ANTEIL ZUR AUSWAHL.

Bilderanalyse: Einsatz und Vergleich von Basic CNN und Transfer Learning inkl. Parametertuning

1. Erkennen von Lungenentzündung (<https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia>)
2. Verkehrszeichen erkennen (<https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign>)
3. Erkennen von Müdigkeit (<https://www.kaggle.com/serenaraju/yawn-eye-dataset-new>)
4. Tierartenerkennung: Erkennung von Katzen oder ... (<https://www.kaggle.com/c/dog-breed-identification> zeigt es am Beispiel Hunde)

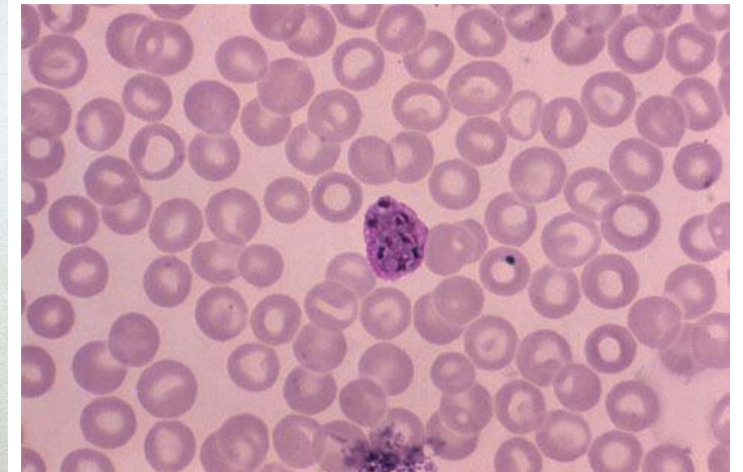
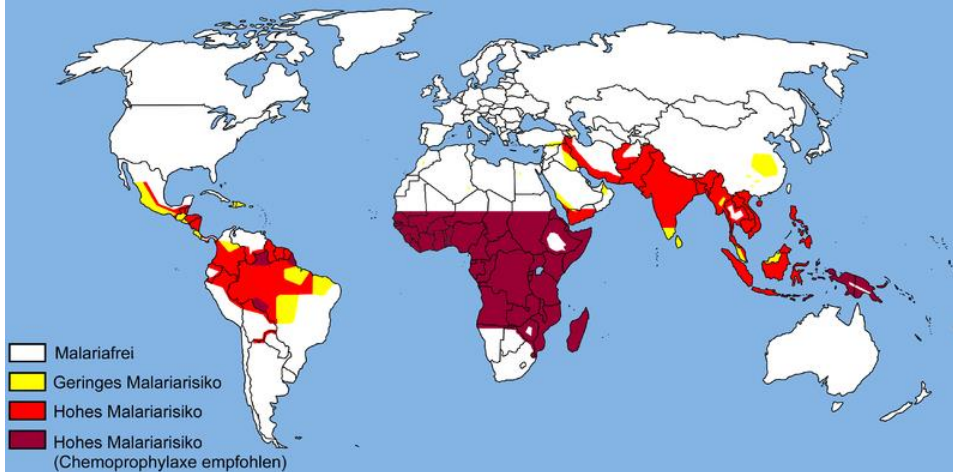
Text Analyse: Einsatz und Vergleich von Basic RNN und LSTM, Attention oder Bert inkl. Parametertuning

1. Fake News Detection/ Classifier mit Deep Learning (<https://www.kaggle.com/hassanamin/textdb3>)
2. Sentiment Analysis of Movies (<https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews>)
3. Börsenkursvorhersage (<https://www.kaggle.com/faressayah/stock-market-analysis-prediction-using-lstm>)

2. FALLBEISPIEL MALARIA

ANWENDUNGSFALL NEURAL NETWORKS: ERKENNEN INFIZIERTER ZELLEN MIT MALARIA.

- Malaria ist eine Infektionskrankheit, die durch den Stich einer Moskito übertragen wird.
 - Häufigste Infektionskrankheit mit ca. 200 Mio. erkrankten Menschen pro Jahr.
 - Diagnose: kostengünstigste Möglichkeit ist Analyse roter Blutkörperchen auf Plasmodien (einzellige Parasiten) per Mikroskop.
- **Aber: wie kann man sowas skalieren auf mehrere Mio. Leute?**



Ziel: Automatisiertes Klassifizieren einer Zelle auf Infiziert oder Gesund per CNN

WIEDERHOLUNG GENERISCHER ABLAUF SUPERVISED LEARNING. ÜBERSICHT.

Notebook mit allen Schritten und
Code liegt auf Ilias und Github

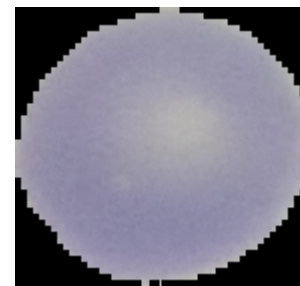
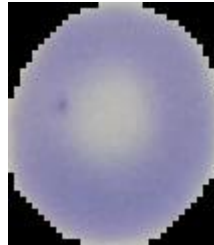
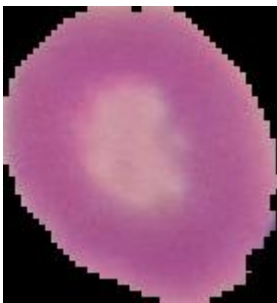
1. Daten organisieren und hochladen.
2. Daten aufbereiten/ Data cleaning.
3. Daten aufteilen in Test- und Trainingsmenge (sowie ggf. Validierungsmenge).
4. Vorbereitungen: Machine Learning Verfahren wählen, Gewichte initialisieren, Kostenfunktion wählen.
5. Training: Schrittweise Optimierung Modellparameter bis Modell möglichst gute Performance für die Trainingsmenge hat.
6. Modell(-güte) validieren anhand der Testmenge.
7. Deployment: Modell einsetzen im „Live“-Betrieb inkl. kontinuierliches Überprüfen Güte Modell und Aktualisierung.

Ziel: Lernen eines möglichst genauen Modells $H(\text{Inputbild}) = \text{Infiziert oder Gesund}$

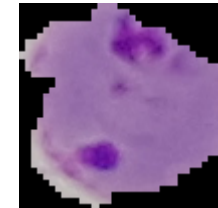
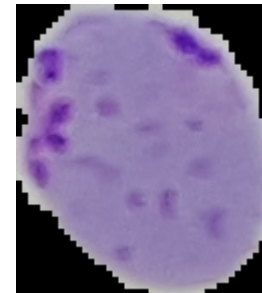
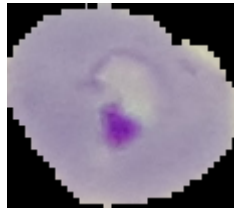
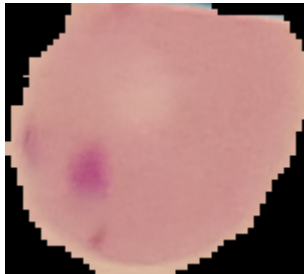
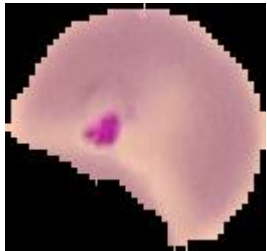
ERKENNEN INFIZIERTER ZELLEN MIT MALARIA.

SCHRITT 1: DATEN ORGANISIEREN.

National Library of Medicine: 27560 Bildern



Gesunde Zellen

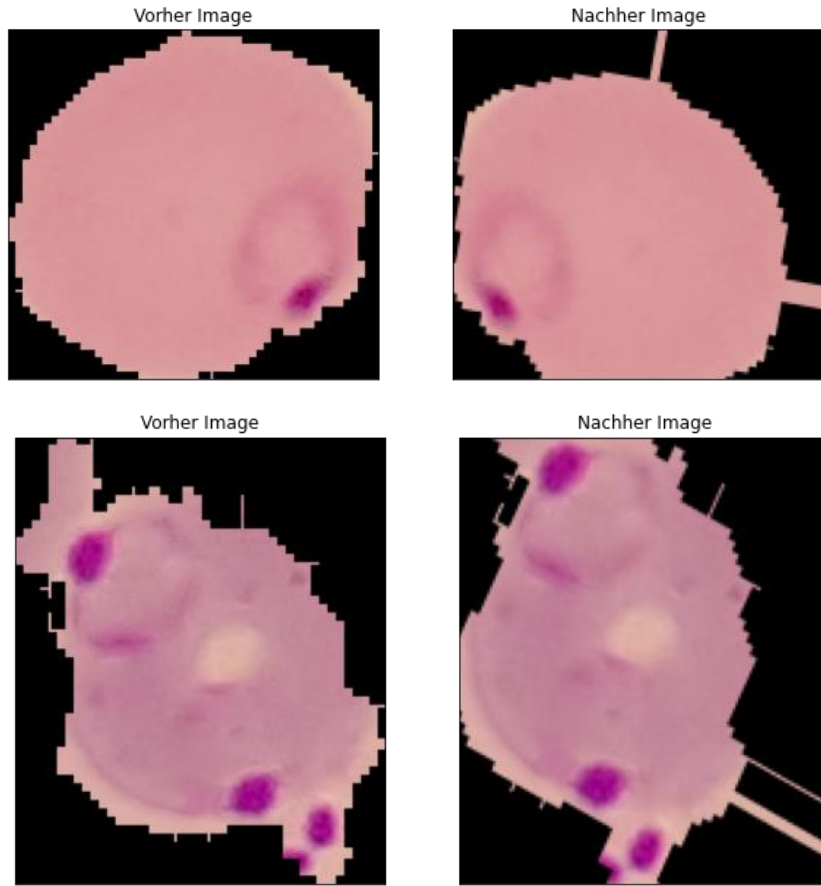


Infizierte Zellen

Wir sehen folgendes:

- Bilder haben verschiedene Größen.
- Zellen haben verschiedene geometrische Formen.
- Grund für Infektion scheint lila Objekt in Zelle zu sein. Dieses kann an verschiedenen Stellen in verschiedener Größe sein.

ERKENNEN INFIZIERTER ZELLEN MIT MALARIA. SCHRITT 2: DATEN AUFBEREITEN/ DATA CLEANING.



Data Augmentation generiert (zufällig) aus vorhandenen Bildern leicht veränderte Bilder als Input für das Modell.

Vorteile:

- Durch die veränderten Bilder wird der Algorithmus gezwungen zu generalisieren, Modell wird somit robuster.
- Durch Data Augmentation kann viel größeres Datenset als vorhanden simuliert werden.
- Data Augmentation geschieht im Arbeitsspeicher.

Nachteile:

- Erhöhter Trainingsaufwand, da mehr Bilder.
- Modell kann Features lernen, die real nicht existieren (verzernte Gesichter, Tiere in bestimmten Farben, ...).

Transfer Learning ist wichtige Methode für das Sicherstellen der Stabilität von Machine Learning Modellen

DATA AUGMENTATION IM DETAIL: AUSGEWÄHLTE OPERATOREN.

Rotation: ImageDataGenerator(rotation_range=90)



Generiert Bilder mit zufällige Rotation um Wert Parameter

Vert. Schieben: ImageDataGenerator(width_shift_range=0.3)



Gen. Bilder mit zuf. Verschiebung links/ rechts < Parameter

Horiz. Schieben: ImageDataGenerator(heighth_shift_range=0.3)



Gen. Bilder mit zuf. Verschiebung oben/ unten < Parameter

Helligkeit: ImageDataGenerator(brightness_range=(0.1, 0.9))



Gen. Bilder mit zuf. Helligkeitswert Parameter; 0.0 = dunkel

HorizontalFlip: ImageDataGenerator(horizontal_flip=True)



Generiert Bilder inkl. zufälligem horiz. Vertauschen

Vertikaler Flip: ImageDataGenerator(vertical_flip=True)



Generiert Bilder inkl. zufälligem vertikal. Vertauschen

Shear Intensity: ImageDataGenerator(shear_range=45.0)



Generiert Bilder durch Stretchen um best. Punkt um X Grad

Zoom: ImageDataGenerator(zoom_range=[0.5, 1.5])



Generiert Bilder per Zoom; < 1 = Vergrößerung



Welche dieser Bilder können in der Realität vorkommen?

ERKENNEN INFIZIERTER ZELLEN MIT MALARIA. SCHRITT 3: DATEN AUFTEILEN IN TEST-, TRAININGS UND VALIDIERUNGSMENGE.

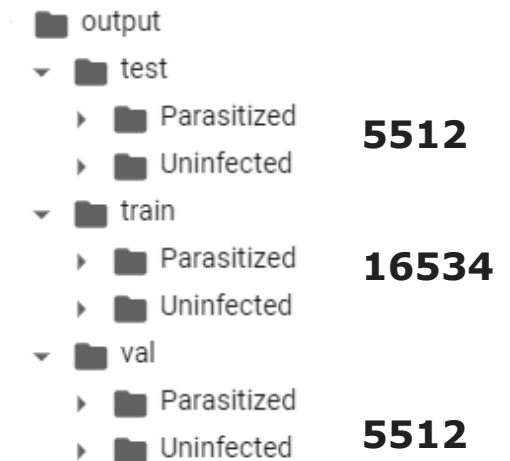
Originaldaten



```
import splitfolders

input_folder = '/content/cell_images'
splitfolders.ratio(input_folder, output="output",
                    seed=1337, ratio=(.6, .2, .2),
                    group_prefix=None)
```

Nach Aufteilung



ERKENNEN INFIZIERTER ZELLEN MIT MALARIA.

SCHRITT 4: MACHINE LEARNING VERFAHREN WÄHLEN.

Simple CNN

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 150, 150, 3)]	0
conv2d (Conv2D)	(None, 150, 150, 32)	896
max_pooling2d (MaxPooling2D)	(None, 75, 75, 32)	0
dropout (Dropout)	(None, 75, 75, 32)	0
conv2d_1 (Conv2D)	(None, 75, 75, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 37, 37, 64)	0
dropout_1 (Dropout)	(None, 37, 37, 64)	0
conv2d_2 (Conv2D)	(None, 37, 37, 128)	73856
max_pooling2d_2 (MaxPooling2D)	(None, 18, 18, 128)	0
flatten (Flatten)	(None, 41472)	0
dense (Dense)	(None, 512)	21234176
dropout_2 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 512)	262656
dropout_3 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 1)	513
Total params: 21,590,593		
Trainable params: 21,590,593		
Non-trainable params: 0		

Transfer Learning mit VGG

Layer (type)	Output Shape	Param #
input_3 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv4 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv4 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv4 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten_1 (Flatten)	(None, 25088)	0
dense_4 (Dense)	(None, 1)	25089
Total params: 20,049,473		
Trainable params: 25,089		
Non-trainable params: 20,024,384		

Transfer Learning mit MobileNet

block14_depthwise (DepthwiseConv2D)	(None, 7, 7, 960)	8640	block14_expand_relu[0][0]
block14_depthwise_BN (BatchNormalizer)	(None, 7, 7, 960)	3840	block14_depthwise[0][0]
block14_depthwise_relu (ReLU)	(None, 7, 7, 960)	0	block14_depthwise_BN[0][0]
block14_project (Conv2D)	(None, 7, 7, 160)	153600	block14_depthwise_relu[0][0]
block14_project_BN (BatchNormalizer)	(None, 7, 7, 160)	640	block14_project[0][0]
block14_add (Add)	(None, 7, 7, 160)	0	block13_project_BN[0][0]
block15_expand (Conv2D)	(None, 7, 7, 960)	153600	block14_project_BN[0][0]
block15_expand_BN (BatchNormalizer)	(None, 7, 7, 960)	3840	block14_add[0][0]
block15_expand_relu (ReLU)	(None, 7, 7, 960)	0	block15_expand[0][0]
block15_depthwise (DepthwiseConv2D)	(None, 7, 7, 960)	8640	block15_expand_relu[0][0]
block15_depthwise_BN (BatchNormalizer)	(None, 7, 7, 960)	3840	block15_depthwise[0][0]
block15_depthwise_relu (ReLU)	(None, 7, 7, 960)	0	block15_depthwise_BN[0][0]
block15_project (Conv2D)	(None, 7, 7, 160)	153600	block15_depthwise_relu[0][0]
block15_project_BN (BatchNormalizer)	(None, 7, 7, 160)	640	block15_project[0][0]
block15_add (Add)	(None, 7, 7, 160)	0	block14_add[0][0]
block16_depthwise_relu (ReLU)	(None, 7, 7, 960)	0	block15_project_BN[0][0]
block16_project (Conv2D)	(None, 7, 7, 320)	307200	block16_depthwise_relu[0][0]
block16_project_BN (BatchNormalizer)	(None, 7, 7, 320)	1280	block16_project[0][0]
conv1 (Conv2D)	(None, 7, 7, 1280)	409600	block16_project_BN[0][0]
conv1_bn (BatchNormalization)	(None, 7, 7, 1280)	5120	conv1[0][0]
out_relu (ReLU)	(None, 7, 7, 1280)	0	conv1_bn[0][0]
flatten_2 (Flatten)	(None, 62720)	0	out_relu[0][0]
dense_4 (Dense)	(None, 1)	62721	flatten_2[0][0]
Total params: 2,320,705			
Trainable params: 62,721			
Non-trainable params: 2,257,984			

Auszug
(zu groß für 1 Screenshot)

ERKENNEN INFIZIERTER ZELLEN MIT MALARIA.

SCHRITT 4: ÜBERSICHT TRANSFER LEARNING VERFAHREN.

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	98 MB	0.749	0.921	25,636,712	-
ResNet101	171 MB	0.764	0.928	44,707,176	-
ResNet152	232 MB	0.766	0.931	60,419,944	-
ResNet50V2	98 MB	0.760	0.930	25,613,800	-
ResNet101V2	171 MB	0.772	0.938	44,675,560	-
ResNet152V2	232 MB	0.780	0.942	60,380,648	-
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.750	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-
NASNetLarge	343 MB	0.825	0.960	88,949,818	-
EfficientNetB0	29 MB	-	-	5,330,571	-

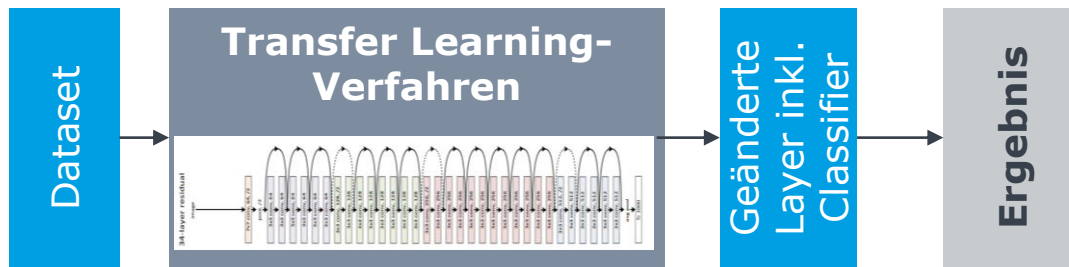
Erklärung Parameter:

- **Size:** Größe des zu speichernden Modells. Besonders relevant für Einsatz in Handys oder IoT-Geräten.
- **Top-1 Accuracy:** gibt an, wie oft im ImageNet-Wettbewerb genau die richtige Klasse erkannt wurde.
- **Top-5 Accuracy:** gibt an wie oft im ImageNet-Wettbewerb eine von 5 gleichzeitigen Vorhersagen des Modells die richtige war.
- **Parameters:** Anzahl Parameter. Relevant falls Modell angepaßt wird, da dies Indikator für benötigte Rechenleistung (und Dauer Training) ist.

ERKENNEN INFIZIERTER ZELLEN MIT MALARIA.

SCHRITT 4: WIE SETZEN WIR TRANSFER LEARNING EIN?

Anpassen/ Ersetzen spezifischer Netzumfänge:



Änderungen notwendig, da wir explizit gesunde von infizierten Zellen unterscheiden wollen.

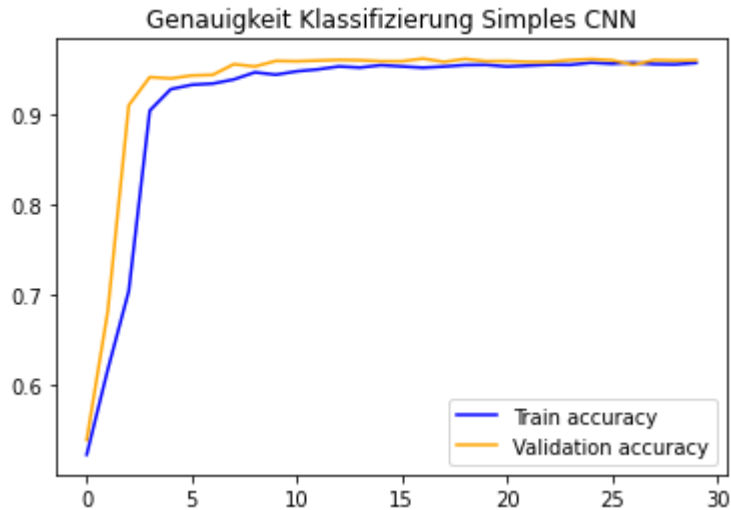
Deshalb müssen wir mindestens die letzte Schicht des Netzes anpassen, in der die Bilder klassifiziert werden.

Wie machen wir das?

- Definition einer Instanz des gewünschten Transfer Learning Modells.
- „Merken“ der letzten Schicht des gewählten Modells.
- Anhängen der gewünschten Layer an die gemerkte, letzte Schicht.
- Modell kompilieren.
- Alle Schichten des Modells auf nicht trainierbar setzen und dann die Hinzugefügten Schichten auf trainierbar setzen.
- Modell trainieren.

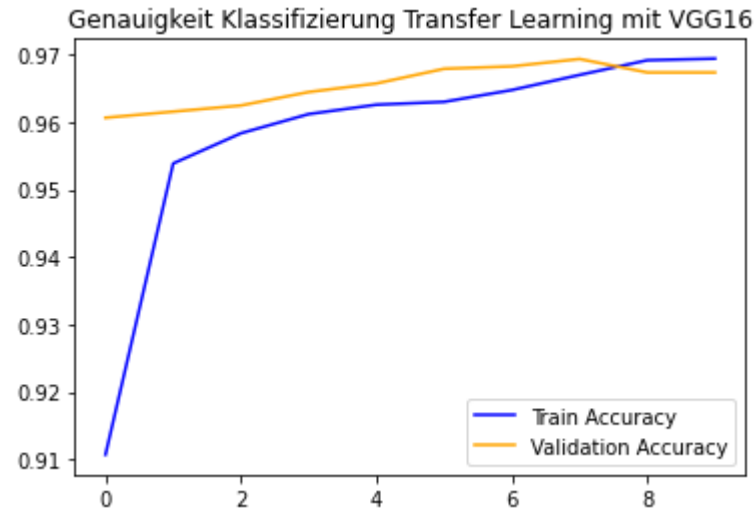
ERKENNEN INFIZIERTER ZELLEN MIT MALARIA. SCHRITT 5: TRAINING.

Simple CNN



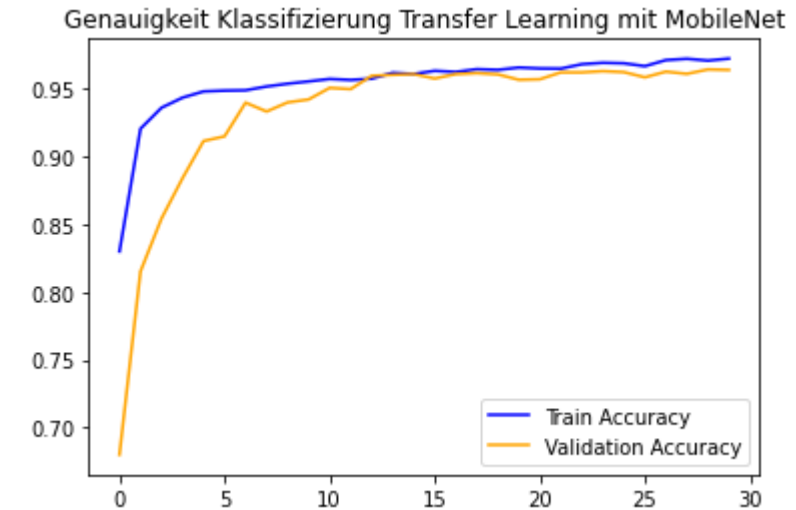
30 Epochen

Transfer Learning mit VGG16



10 Epochen

Transfer Learning mit MobileNet



30 Epochen

ERKENNEN INFIZIERTER ZELLEN MIT MALARIA.

SCHRITT 6: VALIDIERUNG MODELLGÜTE PER TESTDATEN.

- Basic CNN:
 - Accuracy: 0.9554
 - Größe Modell: 247 MB
- VGG16:
 - Accuracy: 0.9623
 - Größe Modell: 153 MB
- MobileNet:
 - Accuracy: 0.9628
 - Größe Modell: 27 MB

CASE STUDY IN GRUPPENARBEIT

Basierend auf dem vorhandenen Notebook [„VL 7 CNN and Transfer Learning - Malaria.ipynb“](#):

1. Data Augmentation:

- Wenden Sie verschiedene Operatoren der Data Augmentation an auf den Data Generator *train_image_gen*.
- Welche der Operatoren, die Sie kennengelernt haben, bilden Zellen ab, die auch in der Realität existieren können?
- Welche Operatoren sind eher nicht geeignet?

2. Simple CNN-Netz:

- Experimentieren Sie mit dem Simple CNN-Netz. Ändern Sie die bestehenden Elemente oder bauen Sie die bekannten Elemente wie Layers, Dropout, Faltungen, MaxPooling etc. ein.
- Welche Elemente ergeben eine höhere Genauigkeit?
- Wie verändert sich die Genauigkeit, falls Sie die Anzahl der Epochen, Learning_rate oder Batch_Size ändern?

3. Transfer Learning:

- Probieren Sie andere Verfahren aus, wie bspw. Inception, ResNet, ...
- Welche ergeben eine höhere Genauigkeit?
- Setzen Sie gerne die Anzahl der Epochen runter, um die Trainingsdauer zu beschleunigen.

LITERATUR UND WEITERE QUELLEN (AUSZUG).

Künstliche Intelligenz:

- Gröner, Heinecke: Kollege KI
- Burkov: The Hundred-Page Machine Learning Book, online verfügbar unter [Link](#)
- Nielsen: Neural Networks and Deep Learning, online verfügbar unter [Link](#)
- Russel, Norvig: Artificial Intelligence – a modern approach
- Produktentwicklung mit AI:
 - Ameisen: Building Machine Learning Powered Applications: Going from Idea to Product
 - Ng: Machine Learning Yearning, online verfügbar unter [Link](#)

Kostenfreie Online-Kurse (bei Interesse):

- Python-Kurse
 - Python for Everybody ([Link](#))
 - Udacity Python Course ([Link](#))
 - **Coursera Course Deep Learning** ([Link](#))
 - FAST AI ([Link](#))

Web-Links:

- CNN: <https://cs231n.github.io/convolutional-networks/>
- CNN: <https://medium.com/machine-learning-researcher/convlutional-neural-network-cnn-2fc4faa7bb63>
- CNN: <https://hackernoon.com/everything-you-need-to-know-about-neural-networks-8988c3ee4491>
- CNN: <https://www.wandb.com/articles/fundamentals-of-neural-networks>