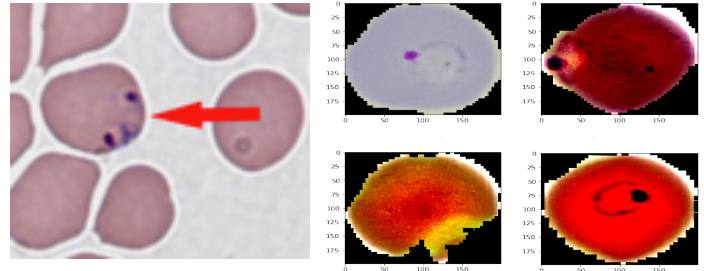


# Malaria Cell Images

Deep Learning Projekt von Pascal Wenger, Vinzenz Frauchiger und Tobias Steinbach

## Beschreibung des Datensatzes

Der Datensatz beinhaltet 27'558 Bilder menschlicher Zellen. 50% davon sind mit dem Malaria Erreger infiziert. Der Datensatz stammt ursprünglich von der U.S. National Library of Medicine. Die Bilder wurden im Chittagong Medical College Hospital, Bangladesh mit einem Android Smartphone mit spezieller App in Verbindung mit einem Standard Lichtmikroskop aufgenommen. Die Bilder stammen von 150 infizierten und 50 nicht infizierten Personen.



## Projektziel

Das Projektziel ist es ein neuronales Netz zu trainieren, welches die infizierten von den nicht infizierten Zellen unterscheiden kann.

## Schritte zur Datenvorbereitung

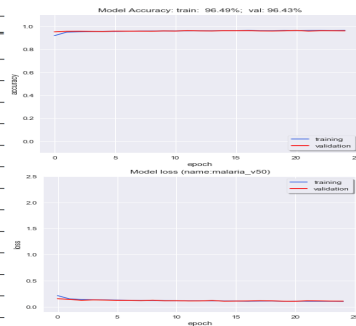
Bevor das Netz mit den Trainingsdaten trainiert werden kann, müssen diese zunächst vorbereitet werden.

- Skalierung der Bilder auf einheitliche Grösse von 128x128x3, respektiv 40x40x3
- Z-Standardisierung der Daten
- Shuffle (zufälliges Mischen) der Daten um ausgewogene Trainings- und Testsets zu erhalten (Stratifizierung nicht notwendig)
- Aufteilung der Daten in Trainings- und Validierungs- und Testdatensatz
- Konvertieren der Y-Werte in One-Hot-Matrix
- On the fly Data Augmentation

## Eingesetzte Netzarchitekturen

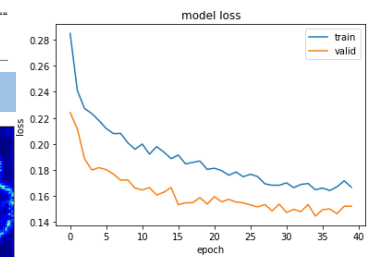
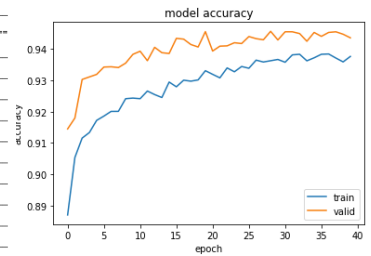
### CNN

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 32, 32, 16)	448
conv2d_2 (Conv2D)	(None, 32, 32, 16)	272
batch_normalization_1 (Batch Normalization)	(None, 32, 32, 16)	64
activation_1 (Activation)	(None, 32, 32, 16)	0
conv2d_3 (Conv2D)	(None, 32, 32, 16)	2320
conv2d_4 (Conv2D)	(None, 32, 32, 16)	272
max_pooling2d_1 (MaxPooling2D)	(None, 16, 16, 16)	0
batch_normalization_2 (Batch Normalization)	(None, 16, 16, 16)	64
activation_2 (Activation)	(None, 16, 16, 16)	0
conv2d_5 (Conv2D)	(None, 16, 16, 24)	3480
conv2d_6 (Conv2D)	(None, 16, 16, 24)	600
max_pooling2d_2 (MaxPooling2D)	(None, 8, 8, 24)	0
batch_normalization_3 (Batch Normalization)	(None, 8, 8, 24)	96
activation_3 (Activation)	(None, 8, 8, 24)	0
conv2d_7 (Conv2D)	(None, 8, 8, 24)	5208
conv2d_8 (Conv2D)	(None, 8, 8, 24)	600
max_pooling2d_3 (MaxPooling2D)	(None, 4, 4, 24)	0
batch_normalization_4 (Batch Normalization)	(None, 4, 4, 24)	96
activation_4 (Activation)	(None, 4, 4, 24)	0
flatten_1 (Flatten)	(None, 384)	0
dense_1 (Dense)	(None, 2)	770
Total params: 14,290		
Trainable params: 14,130		
Non-trainable params: 160		

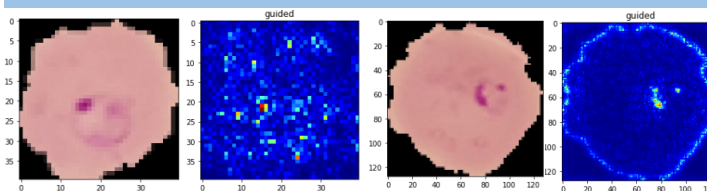


### VGG16

Layer (type)	Output Shape	Param #
vgg16 (Model)	(None, 4, 4, 512)	14714688
flatten_1 (Flatten)	(None, 8192)	0
dropout_1 (Dropout)	(None, 8192)	0
dense_1 (Dense)	(None, 400)	3277200
batch_normalization_1 (Batch Normalization)	(None, 400)	1600
activation_1 (Activation)	(None, 400)	0
dropout_2 (Dropout)	(None, 400)	0
dense_2 (Dense)	(None, 200)	80200
batch_normalization_2 (Batch Normalization)	(None, 200)	800
activation_2 (Activation)	(None, 200)	0
dense_3 (Dense)	(None, 2)	402
activation_3 (Activation)	(None, 2)	0
Total params: 18,074,890		
Trainable params: 3,359,002		
Non-trainable params: 14,715,888		



### Saliency-Maps



	precision	recall	f1-score	support		precision	recall	f1-score	support
0	0.95	0.97	0.96	2713	0	0.94	0.98	0.96	2128
1	0.97	0.95	0.96	2799	1	0.98	0.94	0.96	2162
micro avg	0.96	0.96	0.96	5512	micro avg	0.96	0.96	0.96	4290
macro avg	0.96	0.96	0.96	5512	macro avg	0.96	0.96	0.96	4290
weighted avg	0.96	0.96	0.96	5512	weighted avg	0.96	0.96	0.96	4290

## Schlussfolgerung

Mit sehr einfachen neuronalen Netzen erreicht man bereits einen F1-Score von ca. 93%. Weitere Verbesserungen sind sehr aufwendig. Mit sehr wenig Fachwissen kann bereits sehr gute Ergebnisse erzielen (Keras sei Dank).