1 Einführung

1.1 Problemstellung

Das aktuelle Verfahren zur Analyse von Aufmerksamkeit im Unterricht sieht wie folgt aus. Zyklisch wird immer ein Schüler für eine Minute beobachtet und diese dann bewertet was der Schüler getan hat. Dieses Verhalten wird nun als Gesamtergebnis des Schülers verwendet und als Bewertung des Unterrichtes über die gesamte Klasse gemittelt.
Diese Art der Auswertung ist recht ungenau und Arbeitsintensiv, da sie von einer Person ausgeführt wird. Ein wichtiger Parameter ist die Blickrichtung des einzelnen Schülers, da sie meist dorthin gerichtet ist, wo auch die Aufmerksamkeit liegt.
Ziel ist en nun mit möglichst geringem Aufwand an Hardware eine Bestimmung der Blickrichtung einer ganzen Klasse vorzunehmen. Die Messung soll den Unterricht möglichst wenig beeinträchtigen, wodurch Eye-Tracking Brillen nicht verwendet werden, wegen Kosten und Ablenkung. Auch der Aufbau soll recht einfach und für Laien anwendbar sein, somit wird nur eine festmontierte Kamera vor der Klasse eingesetzt.
Dazu soll ein Verfahren entwickelt werden, mit dem es möglich ist das Filmmaterial von einer gesamte Klasse auf einmal auszuwerten, um von allen die Blickrichtungen während einer Schulstunde zu bestimmen.
[HR92]

- Es sieht für die Beurteilung eines Verhaltens als on- oder off-task drei Dimensionen vor: Blickrichtung, Körperhaltung und Tätigkeit.
  keine simultane Kodierung des unterrichtlichen Kontextes vorgesehen ist und daß über die Validität des Verfahrens nichts bekannt ist.
  von Ehrhardt, Findeisen, Marinello und Reinhartz-Wenzel (1981)

- Münchener Aufmerksamkeitsinventar (MAI)
  Es wird ein festes Zeitintervall von jeweils fünf Sekunden für die Kodierung des Schülerverhaltens und des jeweiligen Unterrichtskontextes vorgegeben.
  vgl. Helmke, 1986

- MAI:
  Die Beobachtungen in einer Unterrichtsstunde umfassen mehrere Durchgänge (SZyklen")
  in Jedem Zyklus alle Schüler in einer vorher festgelegten Reihenfolge für je 5 Sekunden beobachtet und dann Aufmerksamkeits- und Kontextkodierungen.
  Nach jeweils vier vollständigen Zyklen folgt eine zweiminütige Pause

- Die Bedeutung der Aufgaben für das Beteiligungsverhalten der Schüler - Eine Videostudie zur Wirksamkeit des Unterrichtsprozesses
  Zeitintervall $1min$ auf sichtbare einzelne Schüler
  Kodiersystems dienten die Arbeiten von Helmke und Kollegen (Helmke, 1988; Helmke & Renkl, 1992)
  Beobachtungssystemen: molecular composite-Konzept (Hoge, 1985)
  fünf Verhaltensindikatoren bei $\geq 3 \rightarrow$ on task

- Blickkontakt zum legitimen Sprecher oder Objekt
- Aktive Beteiligung an der Aufgabe
- keine Ausübung anderer Tätigkeiten
- keine Motorische Unruhe
- keine Themenferne Kommunikation

1.2 Gesetzte Bedingungen der Anwendung

Damit der Unterricht, wie im Szenario der Problemstellung beschreiben 1.1, möglichst wenig beeinflusst wird, ergeben sich folgende Randbedingungen:

- Brillen, Kontaktlinsen und ähnliches sind erlaubt.
- Die üblichen Bewegungen im Unterricht wie Sprechen, Kopfdrehungen usw. der Schüler ist gestattet.
- Es soll gleichzeitig auf Distanzen zwischen $2.5 - 8m$ zur Kamera auf einer Breit von $6m$ funktionieren.
- Möglichst alle Blickrichtungen der Schüler sollen so exakt wie möglich erfasst werden.

Ein deutsches Klassenzimmer hat $55 - 65m^2$, da noch Abstand zur Tafel usw. beachtet werden muss ergibt sich, wenn die Kamera an der Tafel befindet, einen Abstand zu den Schülern von $2.5 - 8m$ zur Kamera auf einer Breiten von $6m$. Somit muss der Linsenwinkel mindestens $100°$ betragen.
Außerdem soll die Anwendung auf schon vorhanden Aufnehmen eines Unterrichtes arbeiten, bei denen oben genannten Bedingungen erfüllen.

1.2.1 Randbedingungen der Anwendung

Des weiteren werden folgende Annahmen gemacht:

- Die Szene ist Innerhalb eines Gebäudes, mit ausreichend gleichmäßiger Beleuchtung.
- Die Überführung zwischen Welt- und Kamerakoordinatensystem bekannt.
- Die Kamera befindet sich vor der Klasse, so das die Hauptblickrichtung der Schüler in ihrem Fokus liegt.
- Die Gesichter sind komplett sichtbar und nicht verdeckt.

Natürlich sind auch alle inneren Parameter der Kamera bekannt.

1.3 Hardware

Als Messinstrument wird nur eine einzelne Farbkamera eingesetzt. Das Videomaterial der Schulklasse wurde mit einer unbekannten Videokamera aufgezeichnet, daher sind nur die Parameter des Filmes ($640 \times 480??Hz$) bekannt.
Für die Messungen im Versuch wurde die Explorer 4K Action Camera verwendet, sie besitzt eine $170°$ Weitwinkel-Linse mit großem Feald of View. Mit der 2.7K Einstellung wird ein 2688×1520 Video mit 30FPS aufgezeichnet. Sie wurde fest montiert.

1.4 Software

Für die Umsetzung werden folgende Software-Elemente aus fremder Quelle eingesetzt.

1.4.1 ElSe

Ellipse Selection for Robust Pupil Detection in Real-World Environments, ein Algorithmus zur Bestimmung der Pupille im Bild. Der Ursprüngliche ElSe-Algorithmus ist für Graubilder mit Infrarotbeleuchtung ausgelegt, wurde für diese Anwendung aber zu Farbbilder modifiziert.
Entwickelt von der Uni Tübingen. [WF16]

1.4.2 MTCNN Face Detection

Multi-task Cascaded Convolutional Networks, ein Algorithmus zur Detektion von Gesichtern und Bestimmung von 5 Gesichts-Landmarks in Farbbilder. Dabei werden drei CNN auf einer Bildpyramide angewendet um so zuverlässig Gesichter verschiedenster Größe zu erkennen.
[KZ15]

1.4.3 OpenCV

Open Source Computer Vision, ist eine C/C++ Bibliothek von Algorithmen zur Bildverarbeitung in Echtzeit unter der BSD Lizenz (Berkeley Software Distribution)
[Wik15][BK08]

1.4.4 OpenFace

Ein Open-Source Echtzeitverfahren auf Basis von CLNF zur Bestimmung und Analyse von Gesichtsmerkmalen in Grau-Bildern und Videos. Dabei werden 68 signifikante Punkte im Gesicht bestimmt und auf Basis jener Position und Orientierung ermittelt.
Entwickelt von der University of Cambridge [TB16]

2 Theorie & Grundlage

2.1 Gesichtserkennung

Die Gesichtserkennung ist Teil der Bildverarbeitung und wird ständig weiterentwickelt. Darunter fallen neben der Detektion des Gesichtes auch seine Analyse wie Orientierung oder das erkennen von Mimik und Übereinstimmungen.

2.1.1 Künstliches neuronales Netz

Ein künstliches neuronales Netz besteht aus miteinander verknüpften künstlichen Neuronen. Jedes Neuron erhält Eingangswerte, diese erhalten eine individuelle Gewichtung, mittels einer Übertragungsfunktion zusammengefasst und durch eine Schwellenwertfunktion das Ergebnis bestimmt.
Um die Parameter der Neuronen zu bestimmen, werden sie zufällig initialisiert un dann so angepasst,dass sie zu einer gegebenen Eingabe das Gewünschte Ergebnis anzeigt und der Fehler über dem gesamten Trainingsdatensatzes minimal ist.

2.1.2 Convolutional Neural Network (CNN)

Diese ist eine Weiterentwicklung der neuronalen Netzen und werden zur Klassifizierung verwendet unter anderem Im Bereich Bild- und Spracherkennung. Dies wird durch eine gewichtete Faltung erreicht und sind state of the art bei vielen Anwendungen.
So wird die Information aus den umliegenden Punkten eines Bereiches zusammengefasst und komprimiert an die nächste Schicht weitergegeben, um in der untersten Schicht alle vorhanden Informationen zusammenzuführen. Der Faltungskern kann ja nach Anwendung beliebig gestaltet sein, so ist eine Glättung durch einen Gauß-Kernel oder Kantendetektion durch einen Kirsch-Operator möglich.
Ein CNN kann in zwei Bereiche aufgeteilt werden, der Feature Extraktion in welcher durch verschiedene Kernel und Komprimierung die Eingabeinformationen zur Klassifizierung, dem zweiten Bereich, aufbereitet. Gelernt werden können die Kernel an sich und die jeweiligen Bewertungen.

2.1.3 Constrained Local Model (CLM)

Ist ein Verfahren um mehrere Punkte eines Objektes zu lokalisieren. Dabei wird eine Wahrscheinlichkeitskarte für jeden einzelnen erstellt, wo er sich aufhalten kann auf Basis eines Trainingsdatensatzes. Nun wird versucht für das Bild, auf welchem gerechnet werden soll, für jeden Punkt den maximalen Wert zu erreichen zwischen passendem Farbverlauf und Wahrscheinlichkeit.
Dieser Art der Bestimmung von Punkten mit Positionsabhängigkeiten ist ziemlich zuverlässig und dennoch dynamisch genug um auch mit kleinen Veränderungen klar zu kommen.
Dies ist Wichtig, bei der Detektion von verschiedenen Gesichtern in einem Video und zuverlässiger als Active Appearance Model (AAM).

2.1.4 PDM & GAVAM

Mit Point Distribution Model (PDM) können verformbare Objekte recht gut dargestellt werden. Dabei wird die durchschnittliche Form $\overline{X}$ bestimmt und eine Matrix P von Eigenvektoren um die möglichen Deformierungen darzustellen.

$$X = \overline{X} + P \cdot b$$

Somit kann durch einen Skalierungsvektor b alle möglichen Formen X des Objektes dargestellt werden. Zur Vereinfachung reicht es die signifikantesten Eigenvektoren in P auf zu nehmen und dennoch X ausreichend genau zu beschreiben.
Ist bekannt welche Art der Verformung durch den Eingenvektor dargestellt ist, z.B. eine bestimmte Orientierung, so kann anhand des Skallierungsvektors die Rotation des berechneten Objektes bestimmt werden, siehe Generalized Adaptive View-based Appearance Model (GAVAM). Eine Problematik bei dieser Art der Bestimmung der Rotation entsteht, wenn Neben der Verschiebung der Landmarks durch die Rotation, auch eine Deformierung stattgefunden hat und somit niemals eine eindeutige perfekte Lösung gefunden werden kann. Dies ist vor allem de Problematik wenn auf Gesichtern gerechnet werden soll, da immer eine Veränderung der Mundwinkel oder Augenlider vorhanden sind.
[Wik17][Kyb07][MWM08]

2.1.5 Non-maximum suppression (NMS)

Ein Verfahren um Kanten in einem Bild exakter zu bestimmen. Dabei wird der Farbwert des Pixels mit dem umliegenden verglichen und sollte es nicht maximal sein auf Null gesetzt.
Auf diese Weise bleibt nur noch ein Kantenpixel übrig.

2.2 MTCNN Face Detection

Bei Multi-task Cascaded Convolutional Network handelt es sich um ein Verfahren dass bei der Detektion von Gesichtern auch dessen Ausrichtung berücksichtigt um so bessere Ergebnis zu erzielen.

2.2.1 Anforderungen

Sein Einsatzgebiet ist die Vorverarbeitung eines Frames für die spätere Auswertung. Somit muss dieser Schritt von einem möglichst robusten Verfahren zur Detektion von Gesichtern durchgeführt werden.
Dabei wird auf recht großen Bild gearbeitet mit verhältnismäßig kleinen und verschieden großen Gesichtern darin.
Außerdem sollte das Verfahren ausreichend schnell sein, da es sich hierbei nur um ein Vorverarbeitungsschritt handelt und zur Beschleunigung der späteren Berechnung beitragen.

2.2.2 Die 3 Stufen der Verarbeitung

Für die gute Detektionsqualität sorgt die dreistufige Verarbeitung auf der Bildpyramide. Dabei handelt es ich um ein verschieden groß skaliertes Bild, damit der Gesuchte Inhalt in der gewünschten Auflösung abgebildet wird, ohne dass etwas über den Inhalt bekannt sein muss.
Dies ist von Vorteil, damit die CNN auf eine feste Größe von Gesichtern optimiert werden kann, um neben dem möglichen Farbverläufen durch die Skalierung das Lernen zusätzlich zu erschwert.

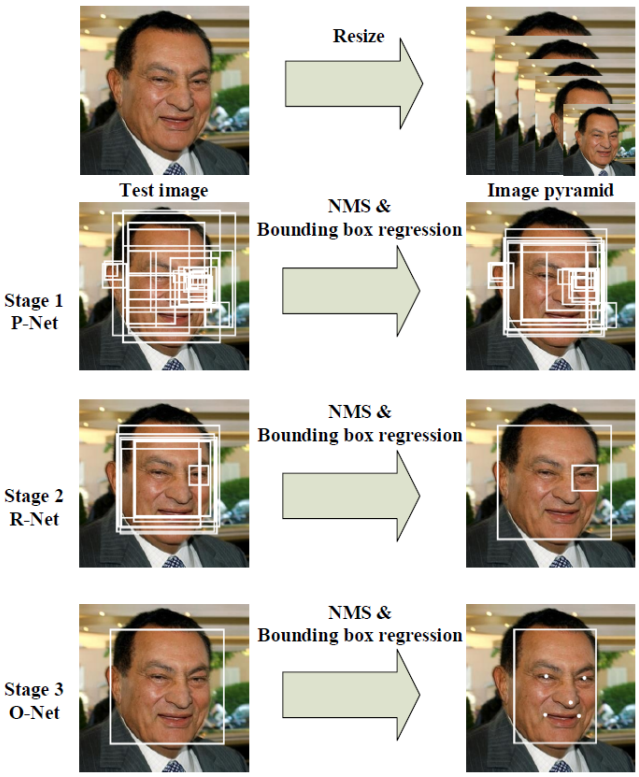

Abbildung 2.1: Darstellung des Funktionsablaufes von MTCCN[KZ15]

Stufe 1

Beim ersten Verarbeitungsschritt erden alle möglichen Bereiche eines Bilds gesucht in denen möglicherweise ein Gesicht zu sehen ist. Dazu wird zuerst ein einfaches CNN eingesetzt und die Ergebnisse, die sich sehr stark überlappen, zusammengefasst.
Für die Detektion wird von einem CNN, dem sogenannte Proposal Network (P-Net), eingesetzt und sehr viele Bounding-Boxen gefunden. Diese werden nun mit einem NMS ausgedünnt, um die am stärksten überlappenden Bounding-Boxen zusammen zu fassen.

Stufe 2

Anschließend werden die möglichen Bereiche mittels eines weiten CNN analysiert, damit alle Nicht-Gesichtsbereiche erkannt und entfernt werden können.
Dies wird von dem Refine Network (R-Net) übernommen und anschließend die möglichen Bounding-Boxen mittels NMS weiter reduziert.

Stufe 3

Der letzte Schritt wird von einem deutlich genaueren CNN übernommen, um ein Gesicht zu detektieren, dem sogenannten Output Network (O-Net). Womit die resultierenden exakten Box und 5 Landmarks ermittelt.

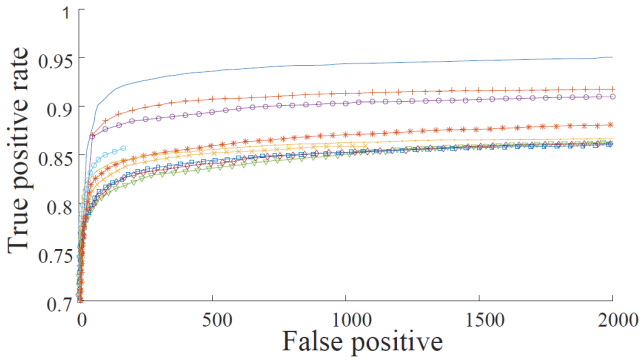

Abbildung 2.2: normale blaue Linie[KZ15]

2.2.3 Qualität

MTCNN Face Detection ist bei der Zuverlässigkeit im Verglich zu anderen bekannten Verfahren überlegen, siehe 2.2. Im Test-Datensatz sind auch Gesichtern mit einer Größe von 20×20 enthalten und wurden erfolgreich erkannt.
Somit sind alle Anforderungen erfüllt um mit diesem Verfahren den vorhanden Frame für die nachfolgenden Berechnungen vorzubereiten, daher wird es auch hier verwendet.

2.2.4 To Do

- Typisches Verhältnis - Frame und Gesichter

2.3 Skalieren von Bildern

Da die Berechnungen meist auf recht kleinen Bildausschnitten ausgeführt wird, müssen diese für weitere Rechenschritte Hochskaliert damit es von OpenFace besser verarbeitet werden kann.
Dabei ist es wichtig, dass die Gesichtsmerkmale möglichst gut rekonstruiert werden, um die entsprechenden Landmarks zu bestimmen.

2.3.1 Nearest-Neighbor

Dieses Verfahren verwendet als neuer Farbwert, den gleichen Wert wie das nächstgelegene Pixel. Dadurch werden nur die ehemaligen Pixel größer und das Gesicht wirkt sehr Kantig, da keine neuen Farbwerte bestimmt werden.

2.3.2 Linear

Dabei wird zwischen den nächst gelegenen umliegenden Pixel linear Interpoliert, wodurch weitere Farbwerte entstehen. Das Ergebnis ist gleichmäßiger als Neares Neighbor, aber immer noch ein recht einfaches Verfahren. Die Kanten werden allerdings unscharf.

2.3.3 Bicubic

Um den Farbwert zu ermitteln, werden die umliegenden 4×4 Pixelwerte betrachtet um den Farbverlauf als eine Funktion 3. Grades zu bestimmen. Somit werden feinere Details besser dargestellt als beim

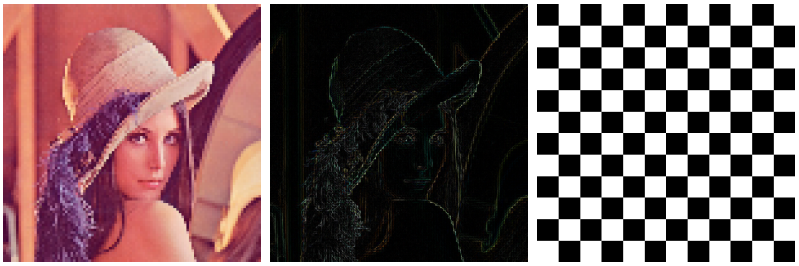

Abbildung 2.3: Die ursprüngliche Abbildung von Lena betrug 100 Pixel Kantenlänge und beim Schachbrett 48 Pixel, beide wurden mittels Nearest-Neighbor auf 512 Pixel skaliert und bei Lena die Differenz bestimmt

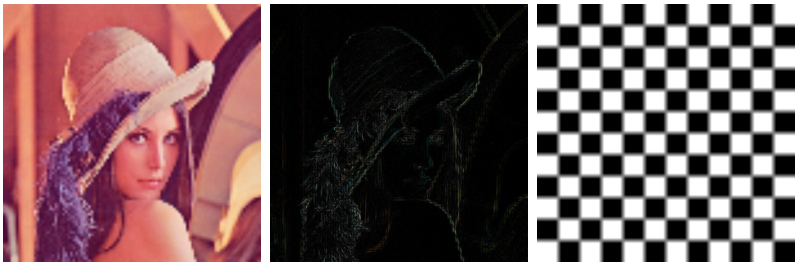

Abbildung 2.4: Die ursprüngliche Abbildung von Lena betrug 100 Pixel Kantenlänge und beim Schachbrett 48 Pixel, beide wurden mittels linearer Interpolation auf 512 Pixel skaliert und bei Lena die Differenz bestimmt

linearen Verfahren und Kanten bleiben stärker erhalten. Allerdings kann es durch den bestimmten Verlauf auch zum Überschwingen kommen, wodurch Fehlfarben entstehen können. [Wik16a]

2.3.4 Lanczos

Dieser Filter besteht aus einer Sinc-Funktion über einen Bereich, um so eine Bewertung der benachbarten Pixelwerte zu erhalten. Somit ergibt sich der neue Farbwert aus den bewerteten umliegenden Pixeln.

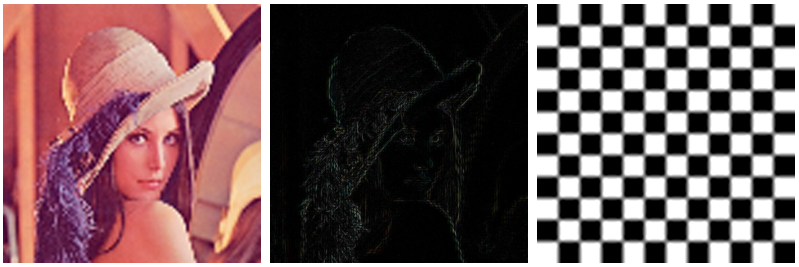

Abbildung 2.5: Die ursprüngliche Abbildung von Lena betrug 100 Pixel Kantenlänge und beim Schachbrett 48 Pixel, beide wurden mittels bikubischem Verfahren auf 512 Pixel skaliert und bei Lena die Differenz bestimmt

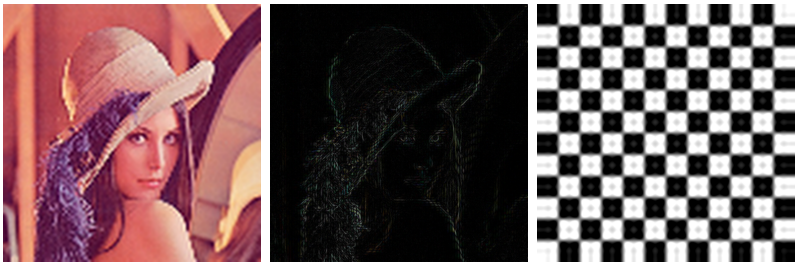

Abbildung 2.6: Die ursprüngliche Abbildung von Lena betrug 100 Pixel Kantenlänge und beim Schachbrett 48 Pixel, beide wurden mittels Lanczus-Verfahren auf 512 Pixel skaliert und bei Lena die Differenz bestimmt

Die Funktion kann und wird für die Anwendung auf einen 8×8 Bereich begrenzt. [Wik16b]

$$L(x) = \begin{cases} \frac{\sin(\pi x)}{\pi x} \cdot \frac{\sin(\pi \frac{x}{a})}{\pi \frac{x}{a}} & \text{wenn } -a < x \geq < a, a \leq 0 \\ 1 & \text{wenn } x = 0 \\ 0 & \text{sonst} \end{cases}$$

2.4 OpenFace

Die Aufgaben von OpenFace ist die Analyse der Gesichtes aus Bildern. Dabei sind nur die Kameraparameter bekannt und keinerlei Zusätze wie eine Tiefenbild oder Infrarotbeleuchtung der Szene. Dabei ist für die Anwendung die Kopfposition (Translation und Orientierung) und Blickrichtung von Interesse, da mit ihnen zurückrechnet werden kann wohin der Schüler schaut.
OpenFace kann neben den Landmarks auch die Position, Blickrichtung und Gesichtsmerkmale bestimmen, basierend auf einem einfachen Bild. Sollte ein Video als Quelle fungieren, so kann OpenFace auch lernen.

2.4.1 Verarbeitungsschritte

Der Rechenaufwand ist so ausgelegt, dass alle Berechnungen auf einer Webcam in Echtzeit ausgeführt werden können, dies ist im aktuellen Fall nicht notwendig, da es sich um eine nachträgliche Auswertung handelt. Durch den Aufbau sind nur recht kleine Farbbilder der Gesichter in einem Video vorhanden wodurch eine Auswertung erschwert wird.

Gesichts-Landmarks: Detektion und Verfolgung

Für die Bestimmung und Tracking der Landmarks wird ein Conditional Local Neural Fields (CLNF) eingesetzt. Dabei Handels es sich im Grunde um ein Constrained Local Model (CLM) nur mit verbesserter Patch Experts und Optimierungsfunktionen.
Zu Beginn werden verschiedene initiale Hypothesen aus dlib-Bibliothek verwendet und die Passende ausgewählt. Bei den initiale Hypothesen handelt es sich um verschiedene Gesichtsorientierungen auf denen verschiedene Netze gelernt wurden. Dies ist zwar langsamer, aber auch exakter als eine einfache Hypothese.
Wird ein Tracing auf Videos gemacht, so wird als initiale Hypothese das Ergebnis aus dem letzten Frame verwendet.

Sollte das Tracing scheitern, so wird das CNN reseted um Neu zu beginnen. Die beiden Hauptkomponenten ist das Point Distribution Model (PDM) zur Erfassung der Anordnung der Landmarks und patch experts zum Erfassen der Variante der einzelnen Landmarks.
Auf diese weise werden 68 Gesichts-Landmarks und weitere 28 pro Auge erfasst. Zur Brechung auf den Gesichtern sollte es eine Mindestbreite von 100 Pixeln für eine zuverlässige Detektion Originalgröße besitzt.

Bestimmung der Gesichtsposition

Zur Bestimmung der Translation und Orientierung des Gesichtes wird ein CLNF eingesetzt. Dabei wurde es mit der Kameraabbildung der 3D-Landmarks eines Kopfes in verschiedenen Positionen initialisiert. Womit auf eine Normierte Abbildung berechnet wird, diese kann mit den passenden Kameraparameter für die Aufnahme angepasst werden um die reale Position zu bestimmen. Sind keine bekannt, so können diese geschätzt werden.
Bei der Schätzung der Brennweite für ein Bild mit einer Dimension $I_x \times I_y$ wird das Standardobjektiv mit 50 mm und einer Auflösung von 640×480 Pixel angenommen, somit ergibt sich für die Brennweiten f_x und f_y:

$$f_x = 500 \cdot \frac{I_x}{640}$$
$$f_y = 500 \cdot \frac{I_y}{480}$$

Bestimmung der Blickrichtung

Durch die Landmarks der Augen werden die Augenlider, Iris und Pupille dargestellt und für jedes Auge separat bestimmt. Dabei wird der Augenbereich, basierend auf dem Gesicht, verwendet um mit einem weiten CNN die 28 Landmarks zu bestimmen.
Zur Bestimmung der Blickrichtung wird wie folgt vorgegeben. Zuerst wird der Strahl bestimmt der, ausgehend vom Zentrum der Kamera, durch das Zentrum der Pupille verläuft. Nun wir der Schnittpunkt zwischen diesem Strahl und einer Sphäre bestimmt, die das Auge repräsentiert. Nun wird ein Strahl bestimmt der vom Zentrum der Sphäre ausgehend durch den berechneten Schnittpunkt verläuft, dies ist die resultierende Blickrichtung.

Detection der Gesichtsmerkmale

Dieser Schritt kann von OpenFace ausgeführt werden, ist aber im aktuellen Fall nicht von Relevanz.

2.4.2 Veröffentlichte Genauigkeit

Die Messung wurde auf de Biwi Kinect head pose dataset und BU dataset. Für die Genauigkeit der Kopfposition haben sich folgend Werte ergeben in Grad:

	Yaw	Pitch	Roll	Mean	Median
Biwi Kinect [FGG11]	7.9	5.6	4.5	6.0	2.6
BU dataset [CSA00]	2.8	3.3	2.3	2.8	2.0
ICT-3DHP [BRM12]	3.6	3.6	3.6	3.6	-

Für die Qualität zur Bestimmung der Blickrichtung ergab sich ein durchschnittlichen Fehler von 9.96 Grad.

2.5 ELSE

Um die Blickrichtung möglichst exakt zu bestimmen, sind die Landmarks der Pupille ausschlaggebend. Zu diesem Zwick kann ElSe eingesetzt werden, da dies ein Verfahren zur Detektion von Pupillen in Bildern unter realen Bedingungen.

2.5.1 Funktion

Das Verfahren ist in der Lage aus Bildern die Umrisse einer Pupille zu ermitteln. Bei realen Aufnahmen sind Bildfehler unvermeidlich, es können Reflektionen (Brille, Kontaktlinse usw.) Make-Up oder körperliche Eigenschaften wie Augenfarbe auftreten und die Detektion erschweren.
Als Ergebnis wird eine Ellipse geliefert als Umriss der Pupille.

2.5.2 Funktionsablauf

Als Input wird im Orginal ein Graubild verwendet, auf dem das Infrarot beleuchtete Auge zeigt. Für den Test wurden Bilder von 384×288 Pixel Größe verwendet und ist auf denen Echtzeit fähig.

Kantendetektion

Da die Pupille als schwarzen Fleck sichtbar ist und die Iris einen helleren Farbton aufweist wird ein Kantendetektor verwendet, der alle Pixel markiert, bei denen eine starke Farbänderung auftritt. Bei ElSe wird ein Morphologischen Ansatz eingesetzt. Von Relevanz sind nur zusammenhängende Kantenpixel, alle anderen können ignoriert werden.

Bestimmen der Ellipse

Um jene Kantenpixel zu erhalten, die die Pupille beschreiben, wird versucht fortlaufende Kanten zu finden, die eine Ellipse bilden, jene die nicht diesen Anforderung entsprechen können recht schnell ignoriert werden. Anschließend können auch alle offenen Ellipsenverläufe verworfen werden und jene die am meisten, vom bestimmten Verlauf abweichen.
Das beste Ergebnis aller so bestimmten, wird als Lösung verwendet.

Grobe Bestimmung

Sollte die Bestimmung der Ellipse scheitern, so wird das Zentrum des dunkelsten Kreises bestimmt, so ein Punkt kann immer gefunden werden, ist aber nicht zwingend die Pupille.

2.5.3 Ergebnisse

Im Vergleich zu en anderen Verfahren im Test, zeig sich das ElSe in den meisten Fällen als Sieger hervorgeht mit einer Verbesserung der Erkennungsrate um 14.53%.
Ein Problem entsteht wenn der Farbunterschied zwischen Iris und Pupille recht gering ist, oder durch Reflektionen der Kantenverlauf gestört wird.

2.5.4 To Do

- alternative Bestimmung genauer

2.6 Berechnung der Position

Zur Bestimmung der Position $P = (X_{avg}; Y_{avg}; Z_{avg})$ des Gesichtes im Kamerakoordinaten wird die Größe, ein Skalierungsfaktor S, des Kopfes im Bild verwendet.
Da bei der Abbildung von den Koordinaten ins Bild gilt $x = f \cdot \frac{X}{Z}$ und $y = f \cdot \frac{Y}{Z}$, somit kann die Tiefe wie folgt abgeschätzt werden.
Sei $P_1 = (X_1; Y_1; Z_1), P_2 = (X_2; Y_2; Z_2)$ die Beschreibung der Größe G eines Kopfes mit.

$$
\begin{aligned}
a &= \frac{\sqrt{(X_1 - X_2)^2 + (Y_1 + Y_2)^2}}{\frac{Z_1 - Z_2}{2}} = \frac{G}{Z_{avg}} \\
S &= \frac{S_G}{G} \\
\Rightarrow a \cdot f &= f \cdot \frac{G}{Z_{avg}} = S_G \\
Z_{avg} &= \frac{f}{S_G} \cdot G = \frac{f}{S} \\
X_{avg} &= \frac{x \cdot Z_{avg}}{f} \\
Y_{avg} &= \frac{y \cdot Z_{avg}}{f}
\end{aligned}
$$

Dies beschreibt allerdings nur eine Annäherung an die Tatsächliche Position, da mit einem Durchschnittlichen Kopfgröße gerechnet wird und je weiter der Kopf vom Zentrum entfernt ist dies mit beachtet werden muss.

2.6.1 To Do

- Bestimmung der Ziele - genauer
  Einbeziehen der Pixelrichtung

3 Implementierung

3.1 Ablauf der Implementierung

Zur Bestimmung der Kopfposition und Orientierung wird ein mehrstufiges Verfahren eingesetzt. Am Anfang müssen alle Gesichter, die im aktuellen Frame vorhanden sind, detektiert werden. Dazu wird die MTCNN Face detection verwendet, da dieses Verfahren auch kleinste Gesichter erkennen kann. 3.2
Für die weiteren Berechnungen muss bekannt sein welchen Bereich das Gesicht in Frame einnimmt und um welches es sich handelt. Der Bereich wird vom MTCNN als Box geliefert, als Personenzuordnung wird ein Matsching zum vorigen Frame verwendet.
Damit auch eine Berechnung auf den kleineren Gesichtern stattfinden kann, werden alle zu kleinen Bildbereiche hochskalliert. Dabei muss wegen Ungenauigkeiten die gefundene Box etwas vergrößert und sollte dann auf eine Mindestgröße gebracht werden. 3.3
Diese Bildbereiche werden nun mit OpenFace weiterverarbeitet, um die Position der Landmarkes im Bild zu bestimmen. Durch die Berechnung auf der selben Person kann das CNN sich auf jeden einzeln einstellen, um so bessere Ergebnisse zu erreichen. Außerdem könne alle gefundenen Personen gleichzeitig (parallel) ausgewertet werden. 3.4
Bei großen Gesichtern wird nun ElSe auf den Augenbereich angewendet, um die Position der Pupille noch exakter zu ermitteln, damit die Blickrichtung genauer wird. Dazu muss allerdings die Differenz zwischen ElSe-Ergebnis und OpenFace-Ergebnis betrachtet werden um Fehler zu erkennen.3.5
Nun wird auf Basis der Landmarks und der Kameraparameter die Position und Orientierung des jeweiligen Gesichtes bestimmt und können für weitere Anwendungen verwendet werden. 3.6

3.2 Detektion der Gesichter

Da nur eine einzige fest montierte Kamera ohne Zoom eingesetzt wird, muss sie eine entsprechend hohe Auflösung besitzen damit alle Personen zu erkenne sind. Allerdings machen die eigentlichen Bereiche der Gesichter nur einen sehr geringen Anteil aus und diese müssen noch Nachbearbeitet werden. Siehe 3.3
Für die automatische Detektion wird Face-MTCNN 2.2 eingesetzt, da dieses Verfahren die meisten Gesichtern mit verscheiden Größen im selben Bild findet, sogar recht kleine mit 20×20 Pixeln. Bei diesem Schritt müssen alle Gesichert gefunden werden, auf denen die Berechnung stattfinden soll. Dabei muss das gesamte Gesicht in der Box sein, ansonsten muss es nicht sehr exakt sein, da OpenFace einen eigenen Facedetector besitzt. Wird MTCNN-Face dedector eingesetzt hat sich eine Vergrößerung der Box um 30% als sinnvoll erwiesen, damit sichergestellt wird, dass alle Merkmale wie Nasenspitzen, Kinn, Augenbrauen usw. sicher im Bildausschnitt enthalten sind.
Ebenfalls in diesem Schritt werden die einzelnen Boxenden Personen zugeordnet, damit im späteren Verlauf das korrekte CNN verwendet wird. Für die Zuordnung reicht meist einen einfache Übereinstimmung der aktuellen Box zum vorigen Frame, da die Gesichter sich meist weder groß Bewegen noch sich die Boxen überlappen.
Damit auf allen Gesichter gerechnet werden kann, Ist eine Semiautomatische Korrektur erforderlich

damit Falsch-Detectionen entfernt und fehlende Boxen ergänzt werden können. Alle nicht gefundenen Gesichtern können manuelle gesetzt oder zwischen dem letzten und nächsten Frame interpoliert werden.
Die gefundenen 5 Landmarks sind für die nachfolgende Berechnung nicht relevant, da sie gerade bei kleinen Gesichtern zu ungenau sind.

3.3 Skalierung auf Mindestgröße

Da OpenFace optimiert ist auf Gesichtern von mindestens 100 Pixel zu arbeiten, werden die Bildbereiche auf diese Größe hochskalliert. 2.3
Diese vergrößerte Box wird nun auf mindestens 130×180 Pixel gebracht, sollte sie kleiner sein. Neben der einfachen Skalierung, muss die Überführung von Bildkoordinaten des Bildausschnittes in die Koordinaten im Frame bekannt sein, damit dies bei späteren Berechnungen berücksichtigt werden kann. Die Skallierung ist für jeden Bildausschnitt individuell und kann sich durchaus über die Zeit ändern, wenn sich z.B. die Distanz zwischen Person und Kamera verändert.
Von einer zu starken Vergrößerung ist abzuraten, da sich dann der Rechenaufwand pro Gesicht erhöht und die Zuverlässigkeit der Berechnungen von OpenFace wieder sinkt.

3.4 Bestimmung der Landmarks

Für die Bestimmung der Landmarks wird OpenFace eingesetzt. Dabei wird jeder Bildausschnitt unabhängig der anderen Betrachtet und da bekannt ist, um welche Person es ich im Bild handelt, kann direkt mit dem jeweiligen CNN gearbeitet werden, das auf diese Person optimiert wurde.
Durch die vorige Selektion wird nun nur auf jenen Bildausschnitten gerechnet auf denen auch die Person zu sehen ist, wodurch die nicht unnötig gesucht werden und auch ein Lernen auf Personen stattfinden kann die nur selten zu sehen sind, da sie nur resettet werden, wenn sie eigendlich zu sehen sein müssten aber nicht detektiert werden.
Für die eigentliche Bestimmung der Landmarks bietet OpenFace zwei verschiedene Methoden, die Berechnung auf Bildern und Videos. Der Hauptunterschied ist das Lernen, dass bei der Videoauswertung verwendet wird, wodurch sich die Bereiche, auf denen Ergebnisse geliefert werden, deutlich erhöht.
Dies ist interessant für die spätere Anwendung, da somit auch Einzelbilder verwendet werden können, die eine deutlich höhere Auflösung haben als ein Video. Allerdings sinkt dann der maximale Winkel relativ zur Kamera beträchtlich, zu Gunsten der maximalen Distanz. Außerdem können schon kleinste Farbänderungen im Bild beim Hochskallieren ausschlaggebend sein, ob ein Gesicht erkannt werden kann, wodurch bei gleicher Bildqualität Gesichter im Video besser erkannt werden.
Da die gesamte Berechnung auf Grau-Bildern basiert ist auch eine Farbkorrektur, wie Verbesserung des Kontrast, Farbverlauf usw. möglich, um etwaige Einflüsse bei der Aufnahme zu korrigieren.
Dennoch kann es passieren, dass trotz allem ein Gesicht falsch detektiert wird, wie z.B. das erkennen eines Gesichtes in der Ohrmuschel, diese müssen entsprechend behandelt werden, da ansonsten das Lernen auf diese Bereiche stattfindend und im nächsten Frame erneut nach diesen Merkmalen gesucht wird.

- Verbesserung durch Farbkorrektur

3.5 Verbesserung der Augen

Zusätzlich zu den 64 Landmarks, die ein Gesicht beschreiben, kann von OpenFace weitere 28 Landmarks für ein Auge bestimmt werden, aus denen dann die Blickrichtung ermittelt wird.
Um die Position der Landmarks zu verbessern, kann auf dem Bildausschnitt der Augen der ElSe-Algotithmus eingesetzt werden. Dieser Algorithmus arbeitet auf einem Farbbild um so die Umrisse der Pupille zu berechnen.
Da unter den 28 Landmarks die Umrisse von Pupille und Iris beschreiben wird, müssen diese aus dem Ergebnis von ElSe abgeleitet werden. Dabei hat sich eine Veränderung des Radius mit ?? für Pupille und ?? für die Iris bewährt.
Allerdings muss das Auge für die Berechnung aus entsprechend vielen Pixeln bestehen, wodurch es im Originalbild mindestens mit 10 Pixeln dargestellt wird, um sinnvolle Ergebnisse zu erhalten. Da diese Berechnung unabhängig der Landmarks ausgeführt wird, empfiehlt sich das Ergebnis zu überprüfen, damit die bestimmte Ellipse auch innerhalb der Augenhöhle liegt.
Dabei wird jedes Auge unabhängig vom anderen betrachtet, wodurch sich verschiedene Blickrichtung ergeben. Ab einer Distanz von mehr als ??cm kann die Blickrichtung beider Augen als parallel angesehen und kann entsprechend behandelt werden. Eine Verbesserung ergibt sich, wenn beide Augen anhängig von einander bestimmt werden, damit sich der Fehler minimiert.

3.5.1 To Do

- Größe für Else
- Grenze für Rechnung

3.6 Bestimmung der Position & Orientierung

Für die Bestimmung der Position und Orientierung des Gesichtes wird wie in 2.6 Beschreiben ausgeführt. Dies kann Wiederrum von OpenFace übernommen werden, dazu muss nur das Zentrum des Bildes und Brennweite f_x, f_y bekannt sein. Außerdem werden noch erweiterte Verfahren angeboten, bei dem die Position im Bild besser mit einbezogen werden, um die Winkel der Kameraabbildung zu berücksichtigen.
Der signifikanteste Parameter für die Position ist die Brennweite f_x, da mit ihm die Tiefe geschätzt wird und sollte entsprechend exakt bestimmt sein. Von Interesse ist vor allem der Punkt auf den der Blick bzw. das Gesicht ausgerichtet ist, dadurch muss neben der Position im Kamerakoordinatensystem auch die Orientierung bekannt sein.
Da nur de Position des Kopfes und seine Orientierung bestimmt werden kann, ergibt sich das Problem, den konkreten Blickpunkt zu ermitteln, da ein ganzer Kegel, wenn eine Fehlertoleranz berücksichtigt wird, die als mögliche Lösungen in Frage kommen.
Außerdem liegt der Blickpunkt meist außerhalb des Bereiches der Kamera und muss entsprechend von einer Anwendung interpretiert werden.

4 Ergebnisse

4.1 Erreichte Werte

- Auswirkung von Pixelrauschen
- Distanzen
- Winkel
- Zuverlässigkeit
- Auswerten der Messung
- Auswirkung der verschiedenen Skalierungesverfahren
  Auswirkung auf die Bereiche
- Wann ELSE
- Mittlung Ergebnis / Landmarks
- Zuverlässigkeit mit Farbkorrektur
- Verschiedene Positionsbestimmungen durch OpenFace

4.2 Fehleranalyse

Mit entsprechend hochauflösenden Kameras können auch bessere Resultate auf größerer Distanz erreicht werden. Gerade die Bestimmung der Blickrichtung ist meist nicht möglich, da die Augenpartie viel zu klein für eine Berechnung ist. So bleibt meist nur die Gesichtsorientierung.
Da Bewegung erlaubt ist passiert es immer wieder, dass Teile des Gesichtes verdeckt werden durch Hände beim Melde, andere Schüler oder dem Lehrer, der vor der Kamera steht oder sich der Kopf zu weit wegdrehen. Aber auch die Frisuren spielen eine Rolle, da dadurch diese einige Landmarks verdeckt werden und so das Gesicht nicht erkannt wird wie z.B. die Augenbrauen.
Eine Lösungsansatz währen Landmarks in Profilbildern zu detektieren und das verwenden von weiteren Kameras aus anderen Perspektiven.

4.3 Verbesserungen

- Mehrere Kameras für 3D und weniger verdecken und wegdrehen

Literaturverzeichnis

[BK08] Gary Bradski and Adrian Kaehler. *Learning OpenCV*. O'Reilly Media Inc., 2008.

[BRM12] Tadas Baltrusaitis, Peter Robinson, and Louis-Philippe Morency. 3d Constrained Local Model for Rigid and Non-Rigid Facial Tracking. In *Computer Vision and Pattern Recognition (CVPR 2012)*, Providence, RI, June 2012.

[CSA00] Marco La Cascia, Stan Sclaroff, and Vassilis Athitsos. Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3d models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(4):322–336, 2000.

[FGG11] Gabriele Fanelli, Juergen Gall, and Luc J. Van Gool. Real time head pose estimation with random regression forests. In *CVPR*, pages 617–624. IEEE Computer Society, 2011.

[HR92] Andreas Helmke and Alexander Renkl. Das Muenchener Aufmerksamkeitsinventar (MAI): Ein Instrument zur systematischen Verhaltensbeobachtung der Schueleraufmerksamkeit im Unterricht. *Diagnostica*, 38(2):130–141, 1992.

[Kyb07] Jan Kybic. Point distribution models, 2007.

[KZ15] Zhifeng Li Yu Qiao Kaipeng Zhang, Zhanpeng Zhang. Joint face detection and alignment using multi-task cascaded convolutional networks, 2015.

[MWM08] Louis-Philippe Morency, Jacob Whitehill, and Javier Movellan. Generalized Adaptive View-based Appearance Model: Integrated Framework for Monocular Head Pose Estimation. In *8th International Conference on Automatic Face and Gesture Recognition*, Amsterdam, The Netherlands, 2008.

[TB16] Louis-Philippe Morency Tadas Baltrušaitis, Peter Robinson. Openface: an open source facial behavior analysis toolkit, 2016.

[WF16] Thomas Kübler Enkelejda Kasneci Wolfgang Fuhl, Thiago C. Santini. Else: Ellipse selection for robust pupil detection in real-world environments, 2016.

[Wik15] Wikipedia. Opencv — wikipedia, die freie enzyklopädie, 2015. [Online; Stand 23. März 2015].

[Wik16a] Wikipedia. Bicubic interpolation — wikipedia, the free encyclopedia, 2016. [Online; accessed 6-May-2017].

[Wik16b] Wikipedia. Lanczos-filter — wikipedia, die freie enzyklopädie, 2016. [Online; Stand 6. Mai 2017].

[Wik17] Wikipedia. Point distribution model — wikipedia, the free encyclopedia, 2017. [Online; accessed 9-May-2017].