

```
0) Get software for JPEG, JPEG2000, JPRG XR and use it for compressing images to a fixed file size (compression ratio).

1) Define face recognition system + dataset and learn to use it.

2) Get software for full-reference and blind (i.e. no-reference) quality metrics and learn to apply it and understand what it does
(3 groups full-reference, 3 groups blind).

3) Find software for Deep - learning based Lossy compression (JPEG AI) and learn to apply it (again: fixed target file size).

4) Choose of of the six new formats and include it into your portfolio (0) + 3))

AVIF (Google, Cisco)

WebP (Google)

HEIC / HEIF (Apple)

JPEG XS

JPEG XL

BPG

5) Apply your face recognition to compressed image data and reevaluate results

6) Prepare a presentation and give it
```

Punkt 0 & 4:

Um unsere Datensätze von .PNG zu .JPEG/.JPEG2000/.JPRG XR und AVIF zu komprimieren wurde weiterhin mit XnConvert gearbeitet. Um allerdings die Bilder automatisch auf die gewünschte Größe zu bringen, haben wir uns selbst ein Python Programm geschrieben, welches nunmehr das manuelle komprimieren überflüssig macht. Für die einzelnen Bildformate ergibt sich eine maximale Komprimierungsrate von bis zu 1:40 für Jpeg und Jpeg2000 und für AVIF und JPEGXR bis zu 1:25.

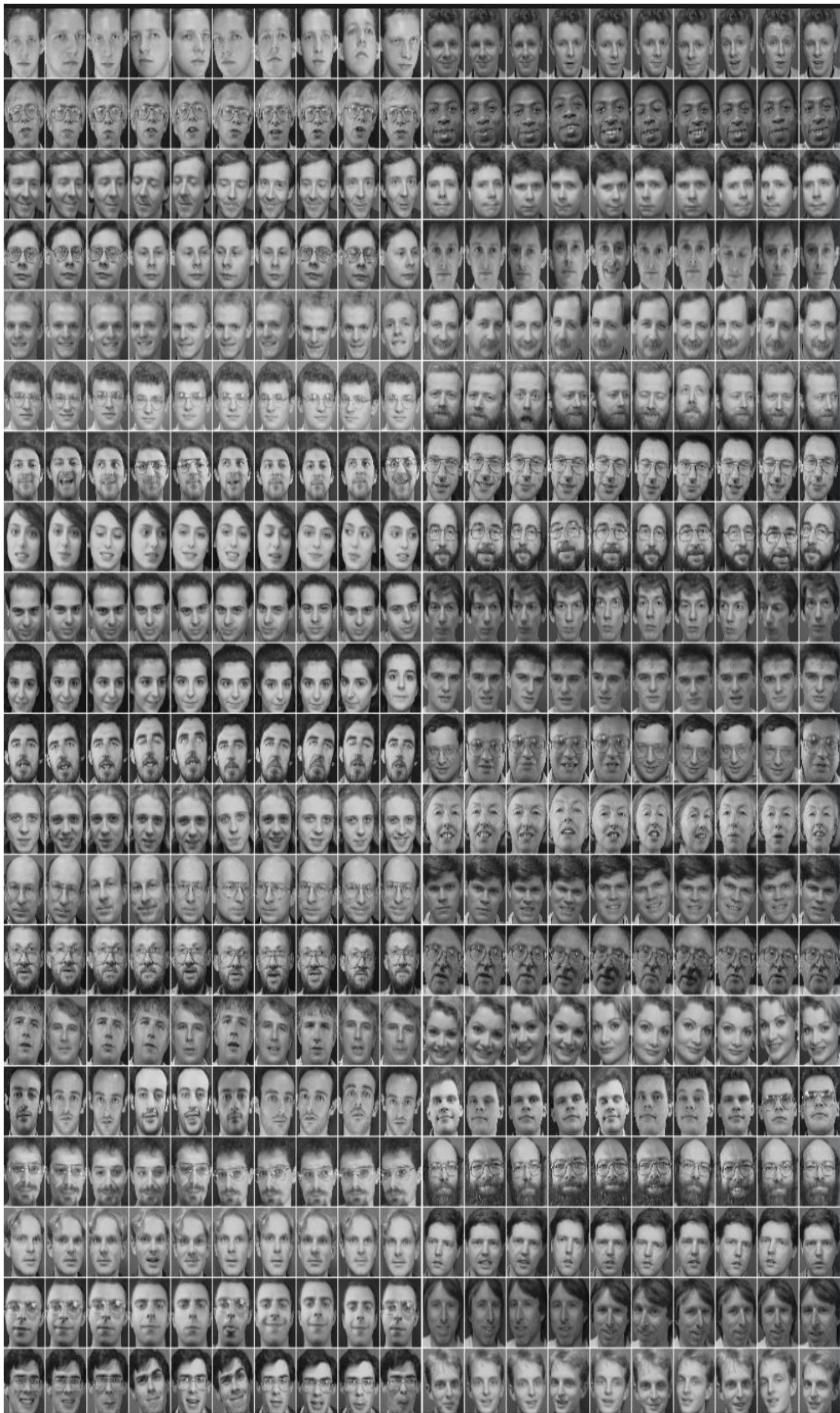
Bei dem von uns gewählten Datensatz, handelt es sich nunmehr um den AT&T-Gesichtsdatensatz¹, welcher früher 'The ORL Database of Faces' genannt wurde. Dieser enthält einen Satz von Gesichtsbildern, die zwischen April 1992 und April 1994 im Labor aufgenommen wurden.

Das Dataset besteht aus 400 Bildern, die eine Größe von 92x112 aufweisen. Innerhalb dieser 400 Bilder finden sich 40 unterschiedliche Personen. Daraus ergibt sich, dass es für jede Personen 10 unterschiedliche Bilder gibt. Diese unterscheiden sich von Lichtverhältnis, Betrachtungswinkel und Gesichtsausdruck. Einen Überblick über das Dataset verschafft das unten beigefügte Bild auf Seite 3.

Alle Bilder haben eine Abmessung von 92x112 und liegen bereits in schwarz-weiß vor.

¹ https://git-disl.github.io/GTDLBench/datasets/att_face_dataset/

Ignaz Ötzlinger
Muhammed Ali Aktas
Vivienne Pesendorfer



Punkt 1:

In dieser Studie wurde die Gesichtserkennung mit den Bildern aus dem AT&T-Datensatz durchgeführt. Dies wurde in Python implementiert und brachte folgende Ergebnisse

Verwendet wurde hier die Eigenface Methode und diese haben wir mit unseren verschiedenen Kompressionsraten getestet.

Es wurde nunmehr angefangen zu schauen, wie genau die Gesichtserkennung bei den einzelnen Komprimierungsraten und den unterschiedlichen Komprimierung Techniken ist.

Bei JPEG mit einer Kompressionsrate von 1:40 war das Ergebnis 90% Accuracy. Im Vergleich bei 1:10 war diese 95%.

Bei JPEG2000 mit 1:40 Rate wurde ein Ergebnis von nur noch 76% erzielt, wobei auch hier bei 1:10 die Genauigkeit noch bei 95% lag.

Bei JPEGXR war bei einer 1:25 Rate das Ergebnis auch knapp über 90% genau wie bei AVIF.

Allgemein kann man sagen, dass die Face Recognition relativ gut mit diesen Raten und Dataset arbeitet. Wobei man gerade bei Jpeg2000 gut sehen kann, dass mit steigender Komprimierungsrate die Genauigkeit abnimmt.

Punkt 2:

full-reference quality metric:

Versucht die Qualität eines Testbildes anhand eines Referenzbildes zu vergleichen, von dem angenommen wird, dass es eine perfekte Qualität hat. Z.b das Originalbild im Vergleich zu einem JPEG komprimierten Bild

Hier haben wir uns für Python entschieden und die von Python zur Verfügung gestellten Funktionen für die full-reference Metric implementiert.²

Verwendete Funktionen:

- **Mean Squared Error (MSE)** misst die durchschnittliche quadratische Different zwischen dem Original und dem komprimierten Bild. Es wird jeder Pixel in der Helligkeitskomponente betrachtet.

Hier gilt umso niedriger desto besser, $MSE = 0$ gibt es nur wenn die Bilder ident sind.

- **Peak Signal-to-Noise Ratio (PSNR)** wird vom MSE abgeleitet und gibt das Verhältnis der maximalen Pixelintensität zur Leistung der Verzerrung an. Hier gilt umso größer der MSE, desto kleiner der PSNR Wert und umso größer der PSNR desto besser.

Typische Werte sind hier zwischen 30 und 50dB und umso höher desto besser.

- **Structural Similarity Index (SSIM)** Hier wird versucht, die Ähnlichkeit der Bildstruktur zu erfassen. Der Begriff „Struktur“ eines Bildes bezeichnet die Abhängigkeiten zwischen benachbarten Bildpunkten, die unabhängig von der Helligkeit und dem Kontrast in dieser Region des Bildes sind.

Hier liegt die Maßzahl zwischen 0 (total unterschiedlich) und 1(identisch)

- **Multi-scale Structural Similarity Index (MS-SSIM)** erweitert den SSIM-Index, indem sie Luminanz Informationen auf der höchsten Auflösungsebene mit Struktur- und Kontrastinformationen in mehreren heruntergerechneten Auflösungen bzw. Skalen kombiniert. Die verschiedenen Skalen berücksichtigen die Variabilität in der Wahrnehmung von Bilddetails, die durch Faktoren wie den Betrachtungsabstand zum Bild, den Abstand der Szene zum Sensor und die Auflösung des Bilderfassungssensors verursacht werden.

*Wenn der Wert näher bei 1 ist, ist die Qualität höher und Richtung 0 sinkt die Qualität.*³

² <https://sewar.readthedocs.io/en/latest/>

³ [https://de.mathworks.com/help/images/image-quality-metrics.html#:~:text=Full-Reference%20Quality%20Metrics,-Full-reference%20algorithms&text=Mean-squared%20error%20\(MSE\),the%20human%20perception%20of%20quality.&text=Peak%20signal-to-noise%20ratio%20\(pSNR\).](https://de.mathworks.com/help/images/image-quality-metrics.html#:~:text=Full-Reference%20Quality%20Metrics,-Full-reference%20algorithms&text=Mean-squared%20error%20(MSE),the%20human%20perception%20of%20quality.&text=Peak%20signal-to-noise%20ratio%20(pSNR).)

```
import cv2
import numpy as np
from sewar import full_ref
from skimage import measure, metrics

#Reference and image to be compared must be of the same size
ref_img = cv2.imread("img1.png", 1)
img = cv2.imread("img2.jpeg", 1)

psnr_sking = metrics.peak_signal_noise_ratio(ref_img, img, data_range=None)
print("PSNR: ", psnr_sking)

from skimage.metrics import structural_similarity as ssim
ssim_sking = ssim(ref_img, img,
                  data_range = img.max() - img.min(),
                  multichannel = True)
print("SSIM: ", ssim_sking)

#default-Werte
msssim_img=full_ref.msssim(ref_img, img, weights=[0.0448, 0.2856, 0.3001, 0.2363, 0.1333], ws=11, K1=0.01, K2=0.03, MAX=None)

print("MSSSIM: ", msssim_img)
```

PSNR: 39.297789509949546
SSIM: 0.9449319774969291
MSSSIM: (0.9879437295852902+0j)

Punkt 3:

Hier haben wir uns für eine **Autoencoder** deep learning compression entschieden. Als Beispiel hätten wir hier:

1. <https://www.tensorflow.org/tutorials/generative/autoencoder>

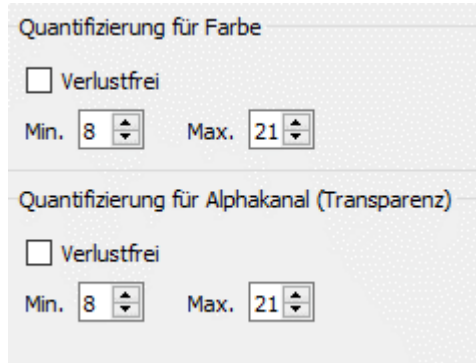
Wir haben auch bereits diesen Code für unsere Dataset Bilder implementiert und können diese auch wieder mit Hilfe des Codes rekonstruieren.

2. https://colab.research.google.com/github/Justin-Tan/high-fidelity-generative-compression/blob/master/assets/HiFIC_torch_colab_demo.ipynb#scrollTo=9BKccvcTpj1k

Hier handelt es sich um den Link, den wir Ihnen bereits letzte Wochen zukommen ließen. Die enthalten Models (hific-lo, hific-mi, hific-hi) könnten verwendet werden um unsere Dataset Bilder auf unterschiedlichen Komprimierungsraten zu bringen.

Punkt 4 AVIF:

Hier wurde unser AT&T Dataset auf das Format AVIF umgewandelt. Für das Format AVIF konnten die 3,30 KB durch das Ausprobieren von den unten gezeigten Parametern erreicht werden:



The image shows a software interface for AVIF quantization. It has two sections: 'Quantifizierung für Farbe' (Color Quantization) and 'Quantifizierung für Alphakanal (Transparenz)' (Alpha Channel Quantization). Each section has a checkbox for 'Verlustfrei' (Lossless), which is currently unchecked. Below each checkbox are two spinners: 'Min.' and 'Max.'. In the 'Farbe' section, the 'Min.' spinner is set to 8 and the 'Max.' spinner is set to 21. In the 'Alphakanal' section, the 'Min.' spinner is also set to 8 and the 'Max.' spinner is set to 21.

Dabei bezieht sich der 2. Parameter auf die Transparenz und der 1. auf die Farb-Quantifizierung selbst.

Punkt 5:

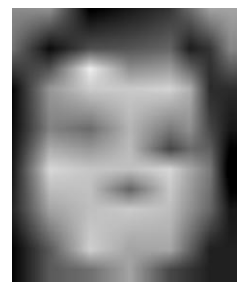
-----NEU-----

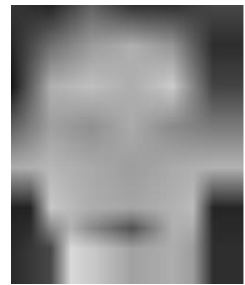
5.0 Beispielbilder:

- **JPEG:**
 - **Komprimierungsrate: links mit 1:25 und rechts mit 1:40.**



- **JEPG2000:**
 - **Komprimierungsrate: links mit 1:25 und rechts mit 1:40.**





- **JEPGXR:**

- Komprimierungsrate 1:25.



- **AVIF:**

- Komprimierungsrate 1:25.



- **Deep L.:**

- Komprimierungsrate 1:25.

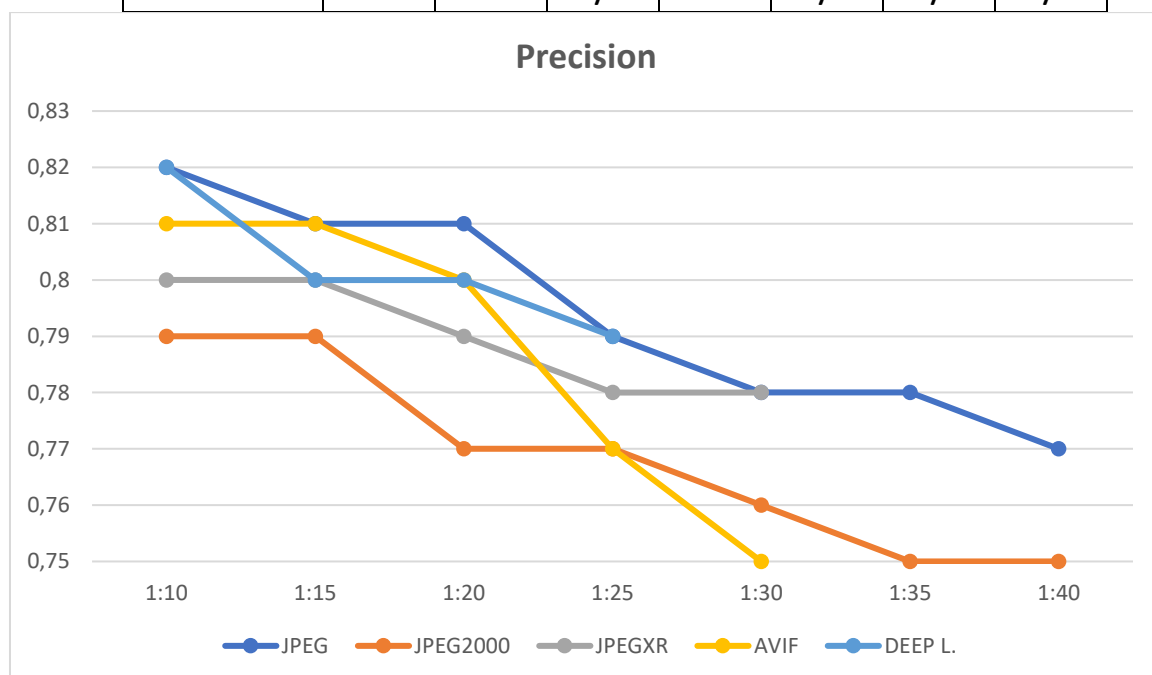


5.1 Ergebnisse

FACE RECOGNITION MIT ORIGINAL BILDERN

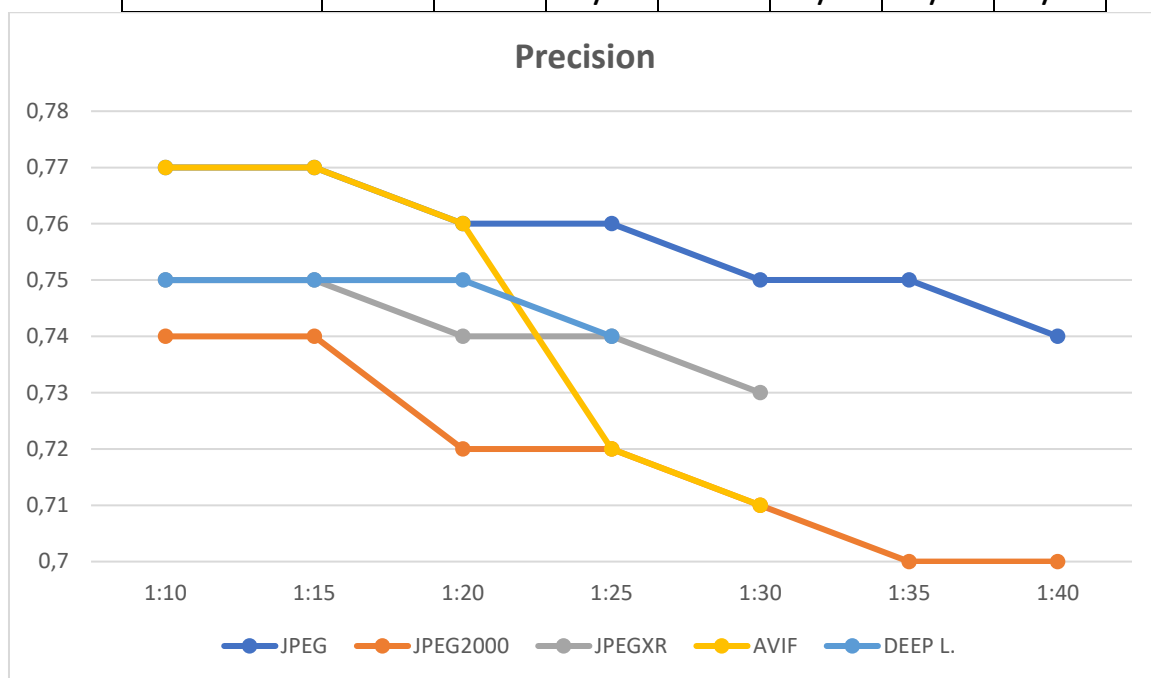
Precision:

	1:10	1:15	1:20	1:25	1:30	1:35	1:40
JPEG	0.82	0.81	0.81	0.79	0.78	0.78	0.77
JPEG2000	0.79	0.79	0.77	0.77	0.76	0.75	0.75
JPEGXR	0.80	0.80	0.79	0.78	0.78	/	/
AVIF	0.81	0.81	0.80	0.77	0.75	/	/
DEEP L.	0.82	0.80	/	0.79	/	/	/



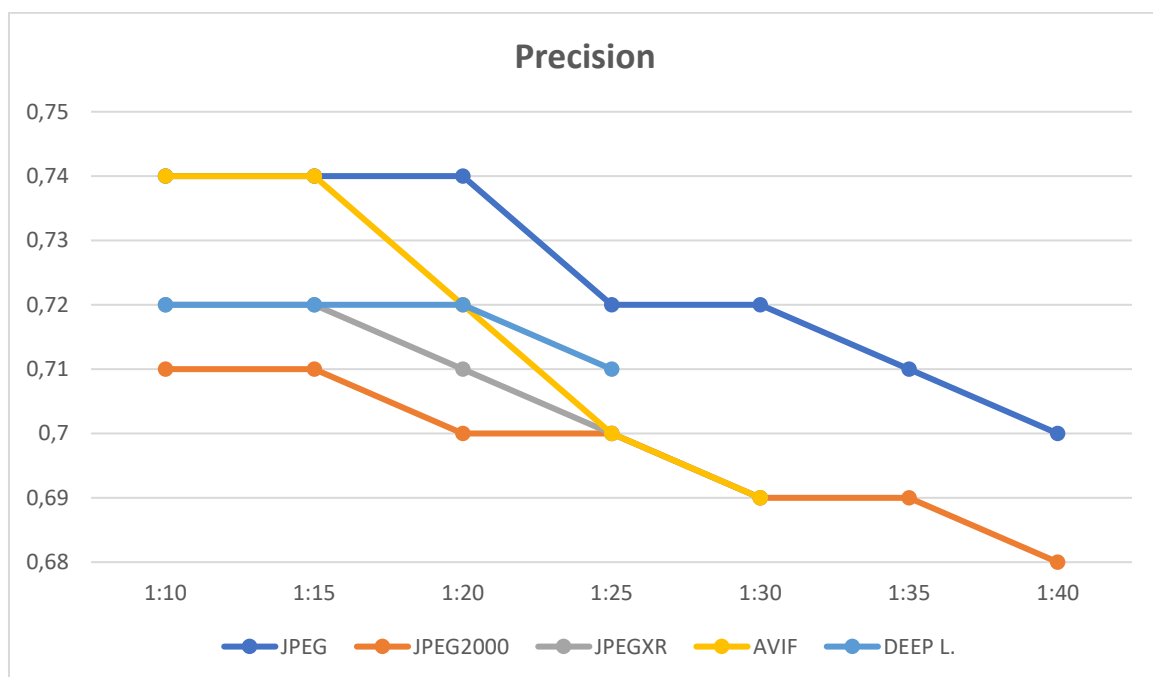
Recall:

	1:10	1:15	1:20	1:25	1:30	1:35	1:40
JPEG	0.77	0.77	0.76	0.76	0.75	0.75	0.74
JPEG2000	0.74	0.74	0.72	0.72	0.71	0.70	0.70
JPEGXR	0.75	0.75	0.74	0.74	0.73	/	/
AVIF	0.77	0.77	0.76	0.72	0.71	/	/
DEEP L.	0.75	0.75	/	0.74	/	/	/



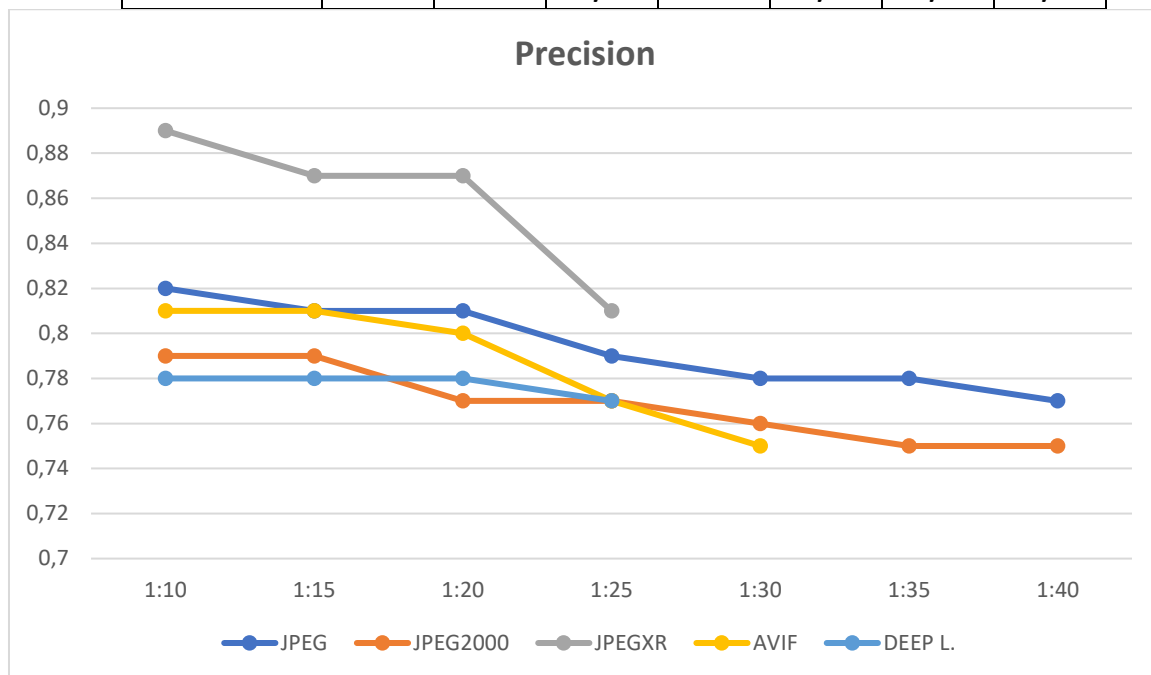
F1-Score:

	1:10	1:15	1:20	1:25	1:30	1:35	1:40
JPEG	0.74	0.74	0.74	0.72	0.72	0.71	0.70
JPEG2000	0.71	0.71	0.70	0.70	0.69	0.69	0.68
JPEGXR	0.72	0.72	0.71	0.70	0.69	/	/
AVIF	0.74	0.74	0.72	0.70	0.69	/	/
DEEP L.	0.73	0.72	/	0.71	/	/	/



Cross Validation (Accuracy Score):

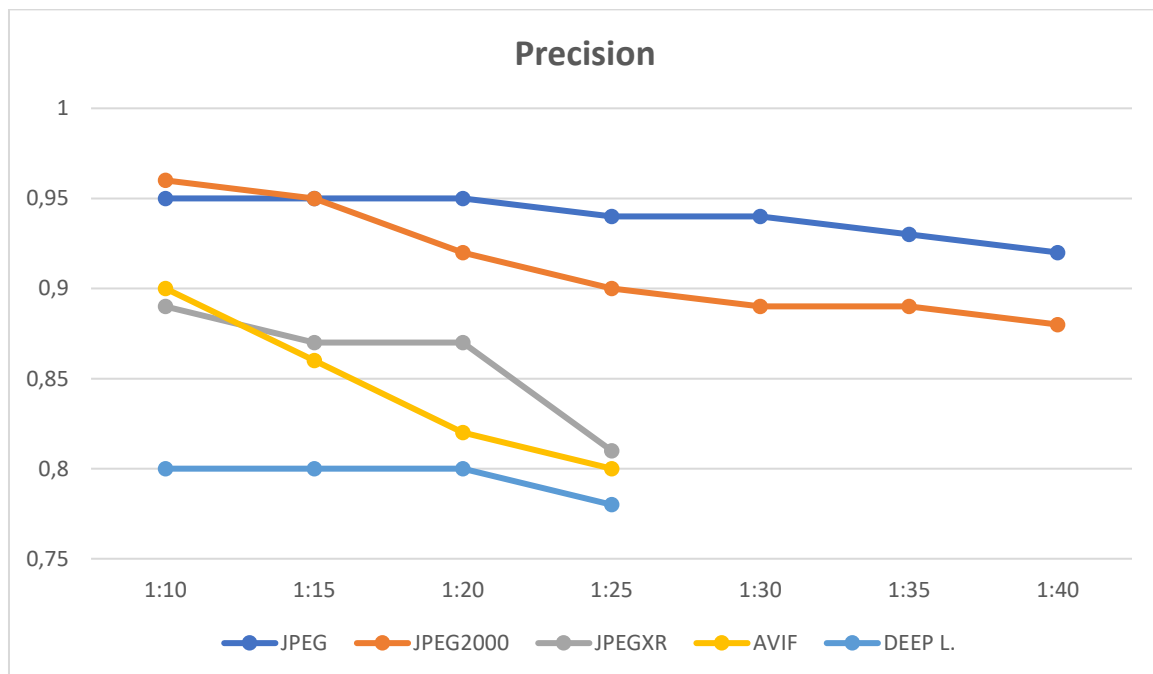
	1:10	1:15	1:20	1:25	1:30	1:35	1:40
JPEG	0.76	0.75	0.75	0.74	0.74	0.74	0.74
JPEG2000	0.78	0.76	0.76	0.76	0.75	0.75	0.74
JPEGXR	0.76	0.76	0.75	0.75	0.74	/	/
AVIF	0.77	0.76	0.75	0.75	0.73	/	/
DEEP L.	0.78	0.78	/	0.77	/	/	/



FACE RECOGNITION MIT KOMPRIMIERTEN BILDERN

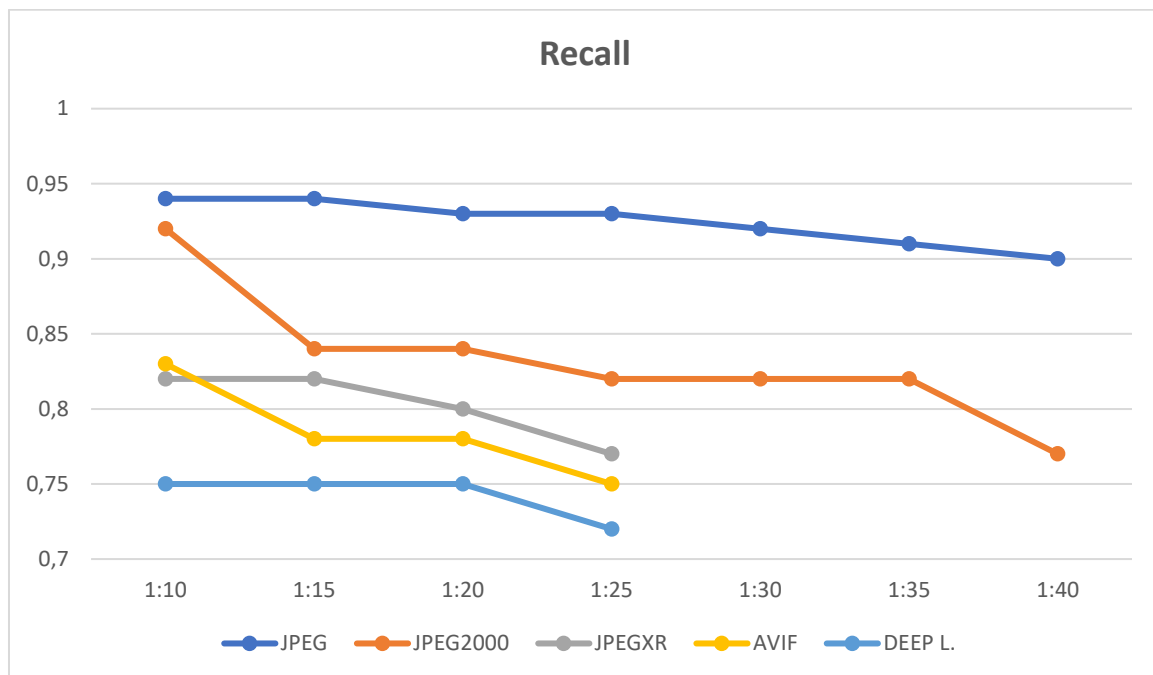
Precision:

	1:10	1:15	1:20	1:25	1:30	1:35	1:40
JPEG	0.95	0.95	0.95	0.94	0.94	0.93	0.92
JPEG2000	0.96	0.95	0.92	0.90	0.89	0.89	0.88
JPEGXR	0.89	0.87	0.87	0.81	/	/	/
AVIF	0.90	0.86	0.82	0.80	/	/	/
DEEP L.	0.80	0.80	/	0.78	/	/	/



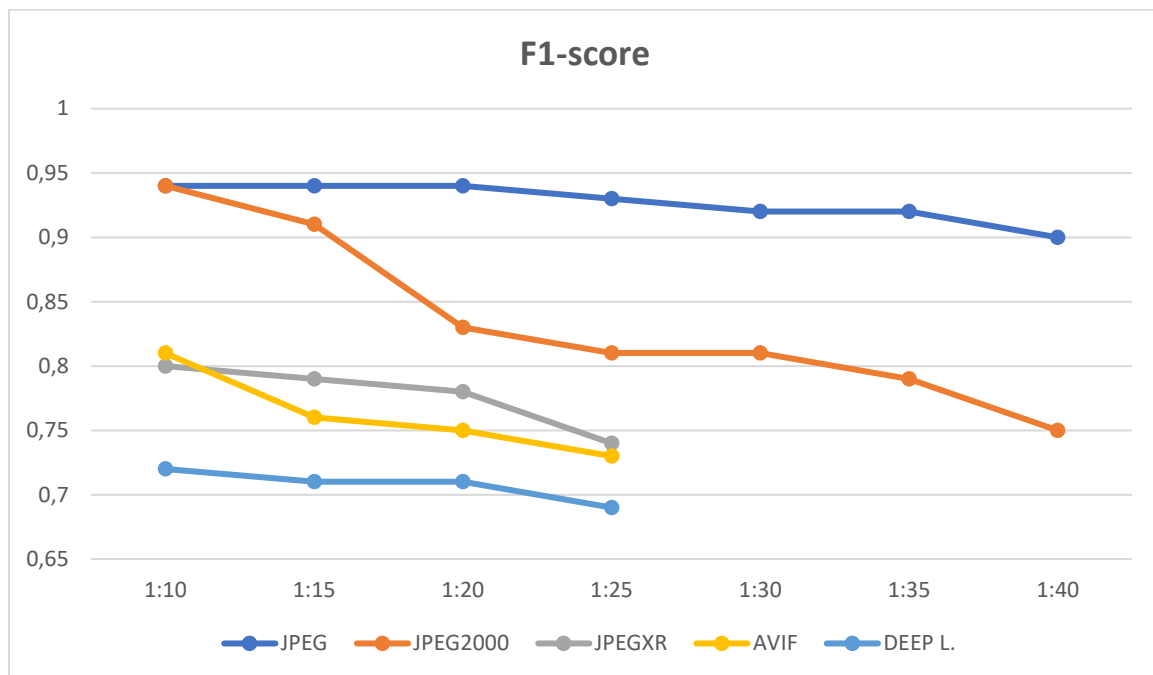
Recall:

	1:10	1:15	1:20	1:25	1:30	1:35	1:40
JPEG	0.94	0.94	0.93	0.93	0.92	0.91	0.90
JPEG2000	0.92	0.84	0.84	0.82	0.82	0.82	0.77
JPEGXR	0.82	0.82	0.80	0.77	/	/	/
AVIF	0.83	0.78	0.78	0.75	/	/	/
DEEP L.	0.75	0.75	/	0.72	/	/	/



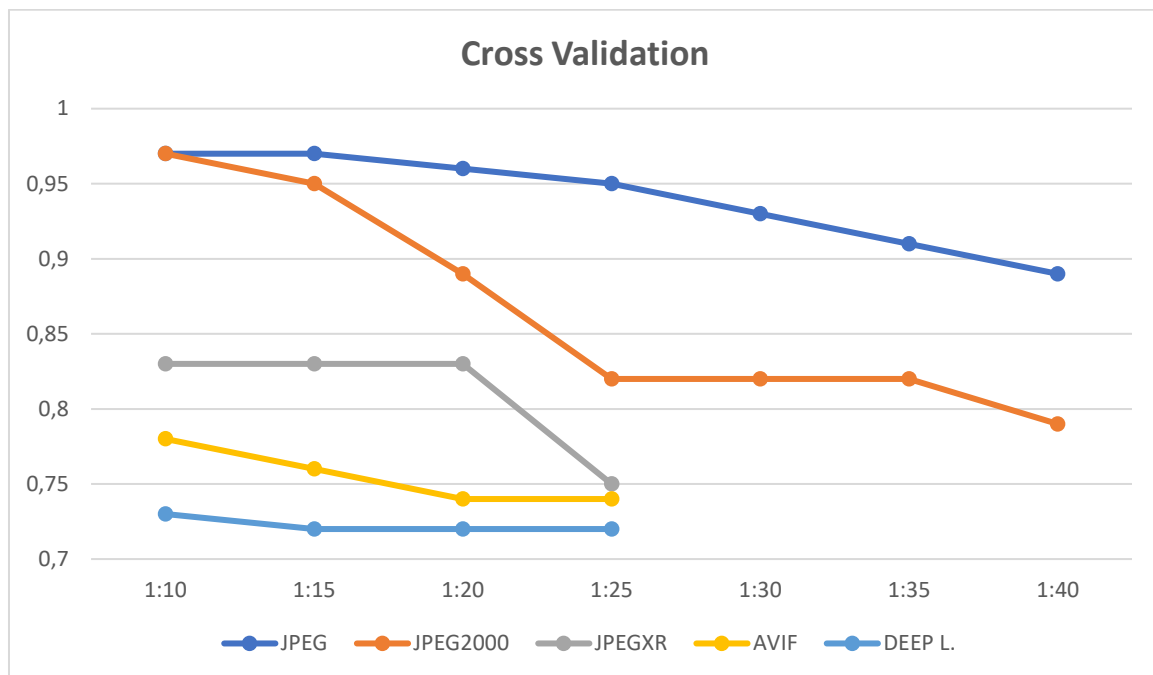
F1-score:

	1:10	1:15	1:20	1:25	1:30	1:35	1:40
JPEG	0.94	0.94	0.94	0.93	0.92	0.92	0.90
JPEG2000	0.94	0.91	0.83	0.81	0.81	0.79	0.75
JPEGXR	0.80	0.79	0.78	0.74	/	/	/
AVIF	0.81	0.76	0.75	0.73	/	/	/
DEEP L.	0.72	0.71	/	0.69	/	/	/



Cross Validation (Accuracy Score):

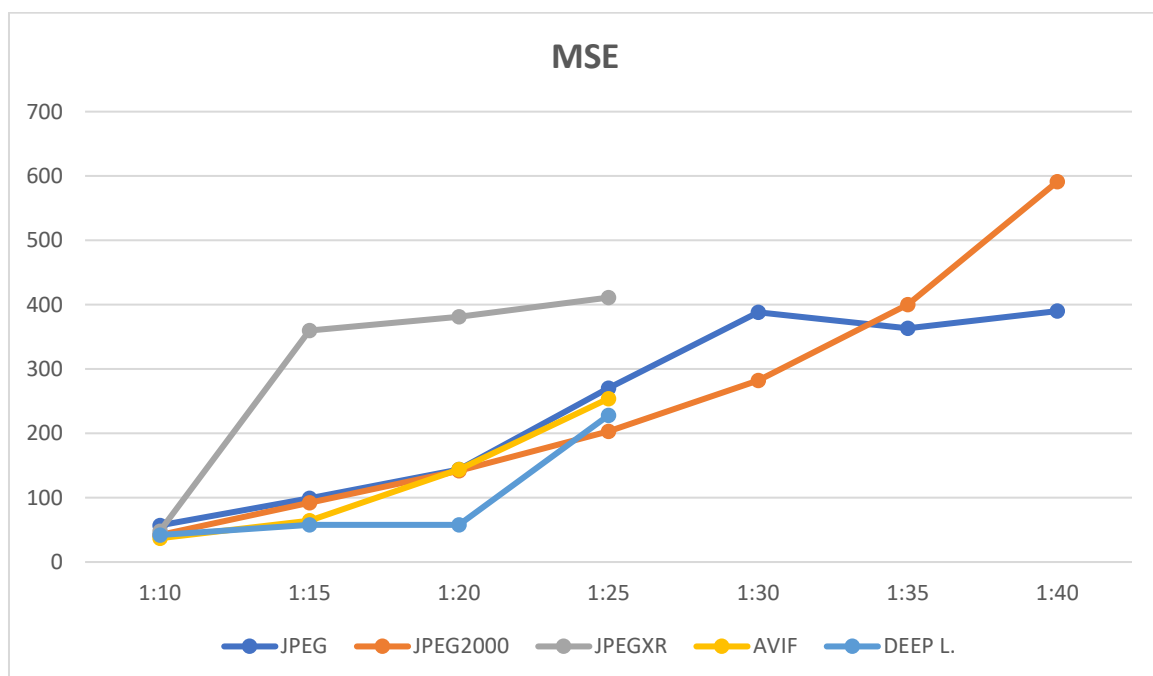
	1:10	1:15	1:20	1:25	1:30	1:35	1:40
JPEG	0.97	0.97	0.96	0.95	0.93	0.91	0.89
JPEG2000	0.97	0.95	0.89	0.82	0.82	0.82	0.79
JPEGXR	0.83	0.83	0.83	0.75	/	/	/
AVIF	0.78	0.76	0.74	0.74	/	/	/
DEEP L.	0.73	0.72	/	0.72	/	/	/

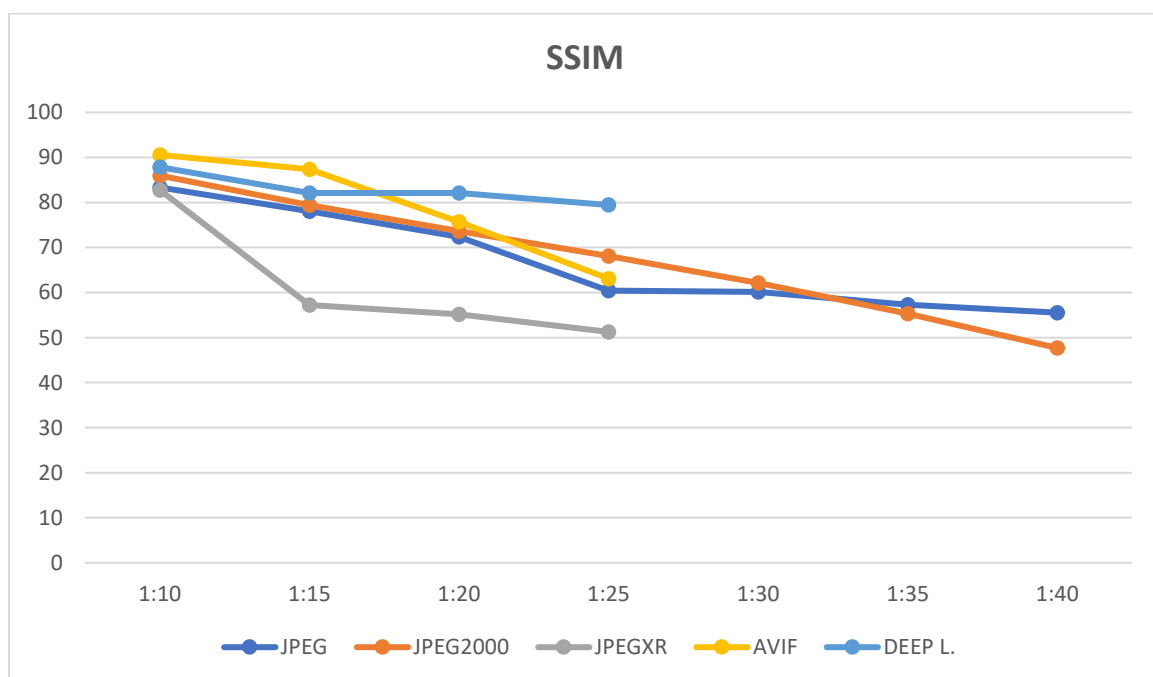
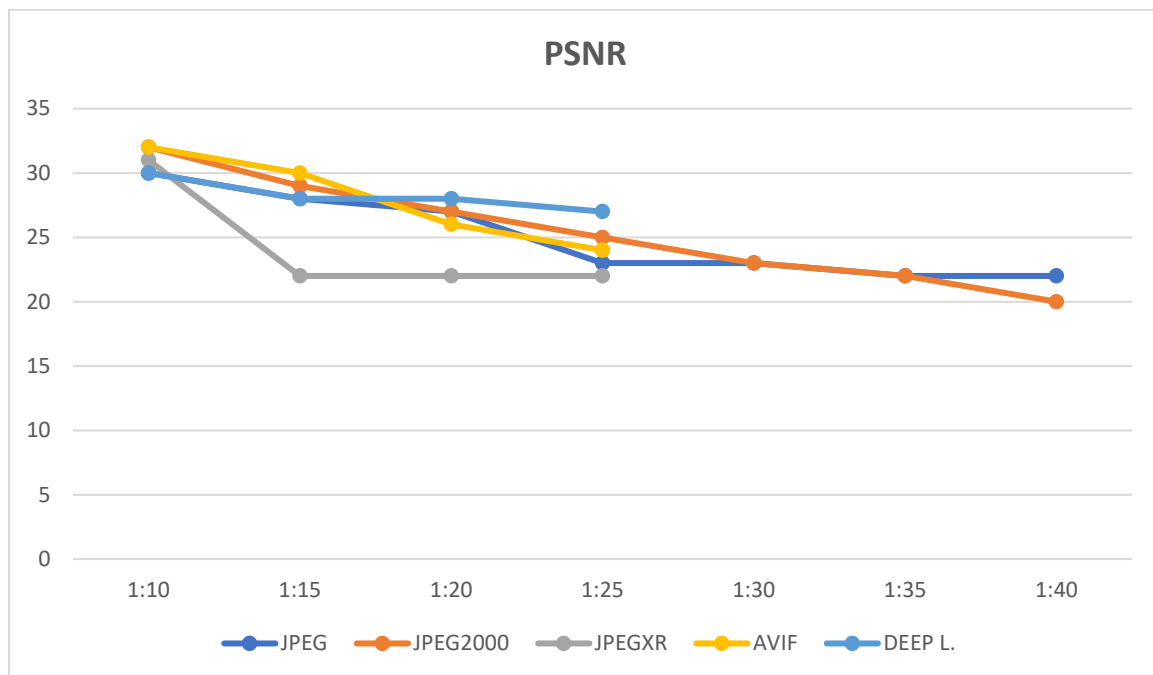


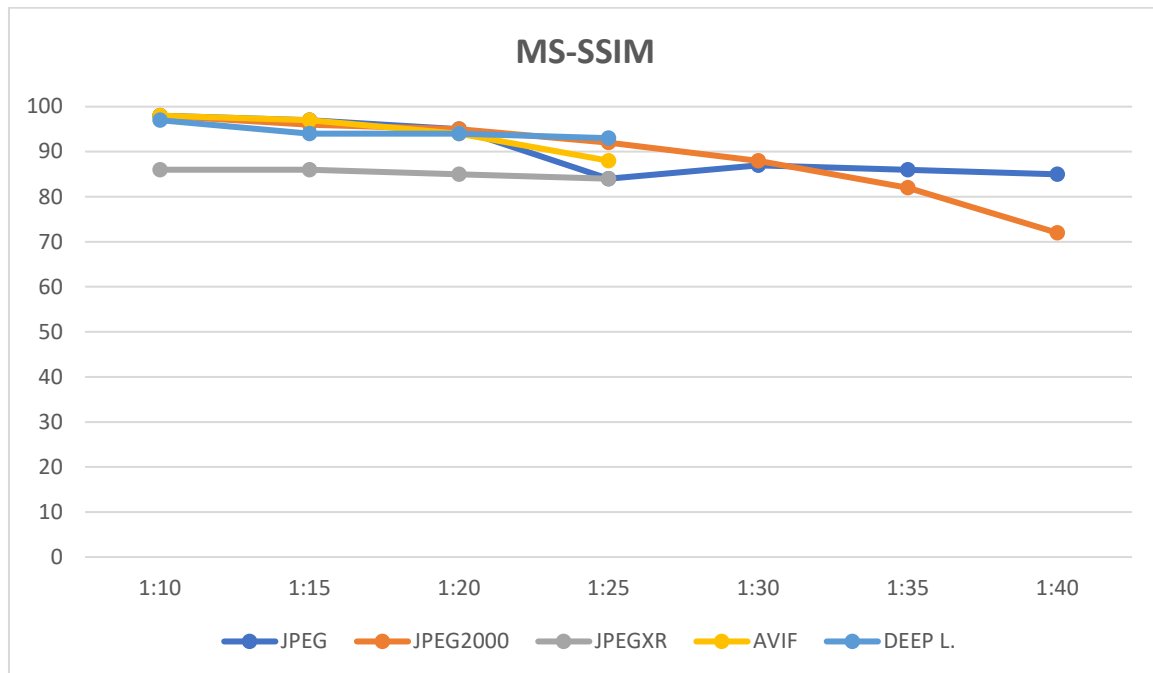
5.3 Full-Reference Quality Metric:

Verwendet wurde der Durchschnitt aus sämtlichen Bildern im Dataset.

Übersicht:







Komprimierungsrate 1:10:

	MSE	PSNR	SSIM	MS-SSIM
JPEG	57.35323903 338512	30.772520080 645222	0.834036235 5012896	0.986512301 8353089
JPEG 2000	42.29010626 9409954	32.236913128 359696	0.859786219 008022	0.985843496 7232376
JPEG XR	48.49880337 732919	31.319233 30980163	0.827666561 0060121	0.869686609 2674722
AVIF	37.47081618 788822	32.471592050 279405	0.905706381 1057786	0.986622969 0272963
DEEP L.	42.33587412 358697	30.145852126 985585	0.878452596 654123	0.975421320 285841

Komprimierungsrate 1:15:

	MSE	PSNR	SSIM	MS-SSIM
JPEG	99.55343701 475162	28.87608000 1420394	0.780848111 846378	0.973798213 8284188
JPEG2 000	92.54567012 81057	29.23119672 1087418	0.794076843 697956	0.969107710 3473705
JPEGX R	360.3462927 018638	22.74160800 2562945	0.572542715 0451871	0.869686609 2674722

AVIF	64.18613079 063142	30.76701058 533209	0.873584962 7033138	0.977720843 1302321
DEEP L.	58.99655487 752502	27.55458792 121821	0.821468821 2315582	0.948744752 2123548

Komprimierungsrate 1:20:

	MSE	PSNR	SSIM	MS-SSIM
JPEG	144.2189343 9440993	27.07041124 9677367	0.723824361 5701379	0.957384872 4424631
JPEG2 000	142.3472418 4782613	27.14956575 2309265	0.736534775 1412736	0.950067137 8862491
JPEG XR	381.8296035 520188	22.42731273 0119347	0.551797599 6138464	0.856357083 4484188
AVIF	144.1670162 072982	26.86861058 186313	0.756938448 6829074	0.943217071 2524208
DEEP L.	/	/	/	/

Komprimierungsrate 1:25:

	MSE	PSNR	SSIM	MS-SSIM
JPEG	270.0870705 551241	23.554334640 944273	0.604328342 7393419	0.847862796 4221974
JPEG2 000	203.6072918 28416	25.451231970 857258	0.680852602 9137475	0.923175884 5059213
JPEG XR	411.5432987 745981	22.254789532 6412457	0.512697519 5438455	0.847720135 4877912
AVIF	254.1790731 7546593	24.267324841 576464	0.630982334 1936772	0.886372637 0549729
DEEP L.	228.0237263 8457553	27.295505595 933868	0.794488873 6968297	0.937520090 1683753

Komprimierungsrate 1:30:

	MSE	PSNR	SSIM	MS-SSIM
JPEG	388.5417236 024856	23.46877432 155535	0.601456556 606354	0.871778251 0716323
JPEG2 000	282.7183132 763973	23.96371601 8709206	0.621512412 736083	0.888505261 272598
JPEGX R	/	/	/	/
AVIF	/	/	/	/
DEEP L.	/	/	/	/

Komprimierungsrate 1:35:

	MSE	PSNR	SSIM	MS-SSIM
JPEG	363.3127513 586954	22.71186215 604556	0.573349311 8799217	0.868849886 2131803
JPEG2 000	400.0728624 805898	22.36072261 4774353	0.553197208 7990357	0.827161308 816631
JPEGX R	/	/	/	/
AVIF	/	/	/	/
DEEP L.	/	/	/	/

Komprimierungsrate 1:40:

	MSE	PSNR	SSIM	MS-SSIM
JPEG	390.7819492 430119	22.32475545 1193997	0.555429951 5834477	0.856482358 0955242
JPEG2 000	591.5362742 624228	20.60949370 6586218	0.477064908 0505909	0.726533521 3314428
JPEGX R	/	/	/	/
AVIF	/	/	/	/

Ignaz Ötzlinger
Muhammed Ali Aktas
Vivienne Pesendorfer

DEEP L.	/	/	/	/
--------------------	---	---	---	---

-----**NEU**-----