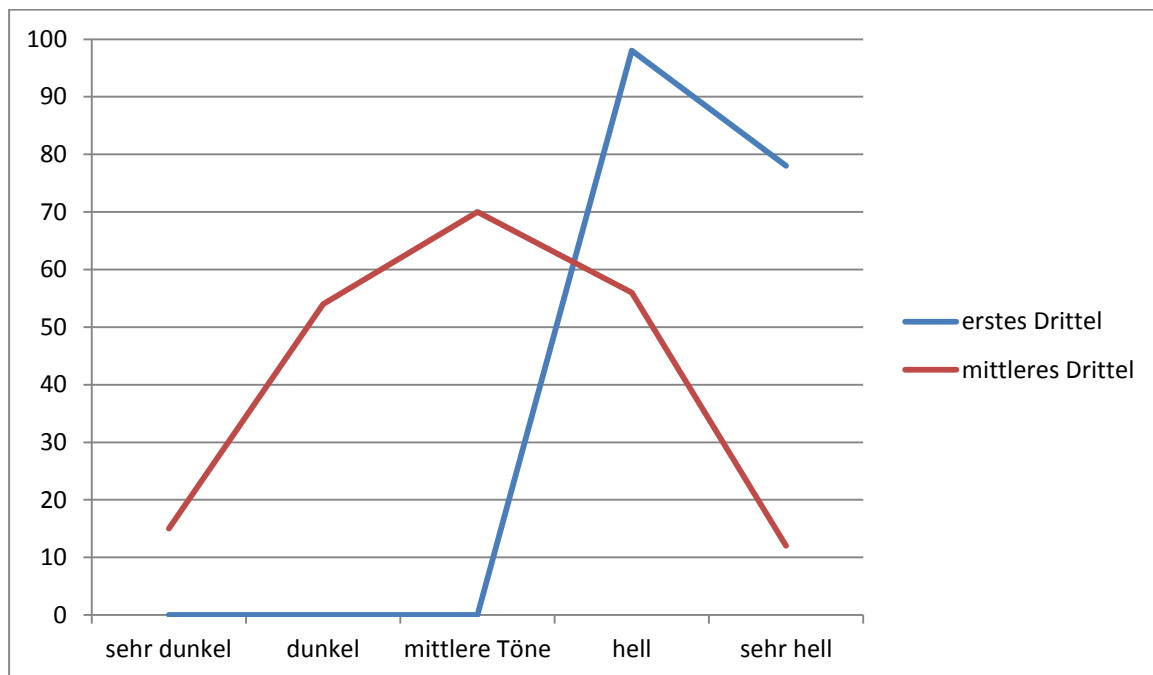


ΔΕΔΙΚΤΕΧΝΙΚ ΒΛΑΤΤ Ξ

Aufgabe 1:

- Ein Histogramm zeigt im allgemeinen wie oft ein Merkmalwert in einem Bild vorkommt. Die x-Achse (bei Bilderhistogrammen) entspricht der Helligkeit/Tonwert, während die y-Achse die Anzahl der Pixel mit diesem Wert zeigt.
- Das Bild hat eine starke Unterbelichtung in dem ersten Drittel (große Anzahl an dunklen Pixeln, während in dem letzten Drittel des Bildes eine Überbelichtung zu betrachten ist (große Anzahl an sehr hellen Tönen)
- Das obere Drittel hat sehr viele helle Töne daher haben wir hohe Werte im Bereich hell- sehr hell, das mittlere Drittel hat viel von allen Tönen, jedoch nicht so viele helle Pixel, daher sieht die Grafik ungefähr so aus.



Aufgabe 2:

- Konvolution - "Faltung" – durch Kombination von dem Farbwert eines Pixels mit den um ihn umliegenden Pixeln wird eine neue Matrix berechnet. Der Farbwert des ursprünglichen Pixels wird dann durch dem neu ausgerechneten Wert ersetzt. Den neuen Wert berechnet man mit Hilfe einer Kernelmatrix zB Blureffekt mit Blurkernel:

Pixel:	3 4 4		Kernel:	1/9 1/9 1/9		3*1/9 + 4*1/9 +4*1/9
	2 12 3	X		1/9 1/9 1/9		= 6
	2 3 3			1/9 1/9 1/9		

Probleme bei den Rändern: wenn wir Pixeln am Rand konvolieren wollen, dann tritt das Problem auf, dass sie teilweise keine umliegenden Pixeln haben, wodurch wir nicht auf 100 % mit der Kernelmatrix Konvolution machen koennen. Man kann daher die Randpixel entweder unveraendert lassen oder das Bild durch wiederholung der Pixel erweitern.

b) Filter 1: Blur von dem Image, da Pixel einen 9 mal kleineren Farbwert nach der Konvolution kriegen.

Filter 2: Sharpen Filter – Bild wird schärfer

Filter 3: Edge Detection Filter – Kanten werden erkannt

Aufgabe 3:

b) **Blur Filter:** ich hab 1/10 für die Konvolution ausgewählt da Pixel einen 9 mal kleineren Farbwert nach der Konvolution kriegen.

Invert Filter: Jeder Pixel hat einen Farbwert, der aus red, blur und green besteht, wobei jede der drei Komponenten einen Wert von 0 bis 255 haben kann. Um die Farben eines Bildes zu invertieren, müssen wir entsprechend jeden dieser Farbwerte passend verändern:

```
int newRed = 255 - red;
int newGreen = 255 - green;
int newBlue = 255 - blue;
```

BW-Threshold Filter:

Threshold benutzt einen spezifischen Thresholdwert, einen min Wert und einen max Wert um die einzelnen Farbkomponenten jedes Pixels entsprechend zu verändern. Farbwerten unterhalb des Thresholdwerts werden mit dem min Wert ersetzt, solche oberhalb mit dem max Wert. Threshold-Wert ungefähr $255/2 \sim 127$ wäre mittelmäßig, d.h wir kriegen ein relativ normales Verhältnis schwarze:weisse Farben im Bild. Bei einem Threshold von 5 zB wäre das Bild fast total schwarz, da zu viele Pixelfarbwerte zu schwarz verändert werden.

Green Filter:

Wir wählen nur den Farbkanal, den wir behalten wollen und setzen die anderen Farbwerten jeweils auf 0, d.h die haben keinen Farbanteil mehr im Bild.

Im Fall Grün:

```
int green = (pixel >> 8) & 0xff;
int targetPixel = green << 8;
```