[](http://www.google.nl/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwiB1-i1upfKAhWBkA8KHX4-DR0QjRwIBw&url=http://akdr.nl/belettering-cijfers-letters/kentekenplaten/gaik-kentekenplaten/&psig=AFQjCNFhrYdr5952C6ywaNz41vSGXKUwJA&ust=1452247702180440)

**Deskewer**De bitmask en de image moeten allebei rechtgezet worden. Hiervoor hebben we een functie deskewerImage(Image,C) gemaakt waarbij C de huidige hoekpunten zijn. We gebruiken de functies fitgeotrans en imwarp om een geometrische transformatie uit te voeren op de image. ***Plaatje van rechtzetten***

Als de kentekenplaat eenmaal goed geroteerd is dan moet de afbeelding zo geknipt worden dat je alleen maar de kentekenplaat overhoudt. Hiervoor hebben we de functie cutInCorners(Image, Corners) gemaakt waarbij corners de hoekpunten zijn van de rechtgezette bitmask. De functie zegt dat je als imageout alleen maar datgene geeft dat binnen de 4 hoekpunten ligt. ***Plaatje van uitgeknipte plaat!***

**Volgende stappen**Nu we de kentekenplaat uit de video hebben weten te halen is het tijd om ook echt aan letterherkenning te gaan beginnen. De stappen die we willen gaan doorlopen is om de objecten te gaan scheiden van elkaar en dan met neural networks de letters te gaan herkennen.

**GetPlateBitmask**GetPlateBitmask is het filter dat we gaan gebruiken om de gele nummerplaat beter te kunnen herkennen. Rood is te zien op kanaal 1, blauw op kanaal 2 en groen op kanaal 3 zoals we met het masker van de colorThresholder hebben uitgerekend. Hierdoor zul je dus, als je dit masker toepast op het oorspronkelijke plaatje, een afbeelding terugkrijgen met de kentekenplaat (in rood gemarkeerd, **zie plaatje**) en ook nog wat ruis.

Uiteindelijk zullen we ons bitmasker voor de locatie van de kentekenplaat dus als volgt verkrijgen: bm= createMask(normalize(data)). Op het gethresholde bitmasker passen we als laatst een closing toe en selecteren we het grootste object zodat we geen ruis meer overhouden.

**Hoekpunten**Voor de rotatie van de kentekenplaat is het belangrijk om eerst de cornerpoints te vinden. Hiervoor hebben we een functie findCorners(bitmask) gemaakt waarmee we de 4 hoeken berekenen en deze teruggeven. In de originele video worden d.m.v. deze hoeken de kentekenplaten gemarkeerd (zie de rode omlijning). ***Plaatje van markering (in de GUI moet dit)***

**De opdracht**We hebben de opdracht gekregen om met behulp van beeldverwerking een programma te maken dat kentekenplaatnummers herkent. Dit programma is gemaakt met Matlab en zal o.a gebruik maken van segmentatie, edge detection, object separation en neural networks.

**GUI**In de GUI is de video te zien waarop je kentekenplaatherkenning gaat toepassen. Je kunt een video inladen en deze starten, stoppen of het volgende frame laten zien m.b.v. 3 knoppen. Ook zie je op de video een rode rand om de kentekenplaat heen.

**Create Mask**We hebben een stuk code gegenereerd met behulp van colorThresholder dat een masker maakt. Dit hebben we gedaan door een beeld te geven aan de colorThresholer en alleen de RGB waarden over te houden die de gele kentekenplaat bevat.

Kentekenplaatherkenning

Kentekenplaatherkenning