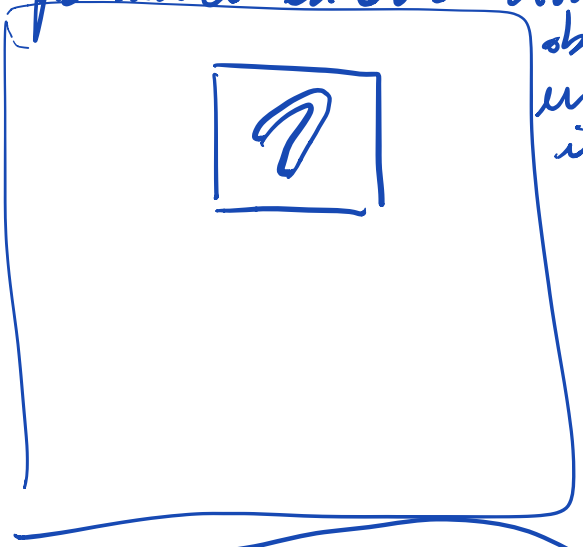
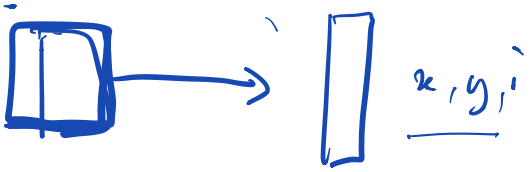
Stage B : Définition sujet + priorités

pb usuel en CV : trouver et classifier un objet dans une grande image.



x, y, i

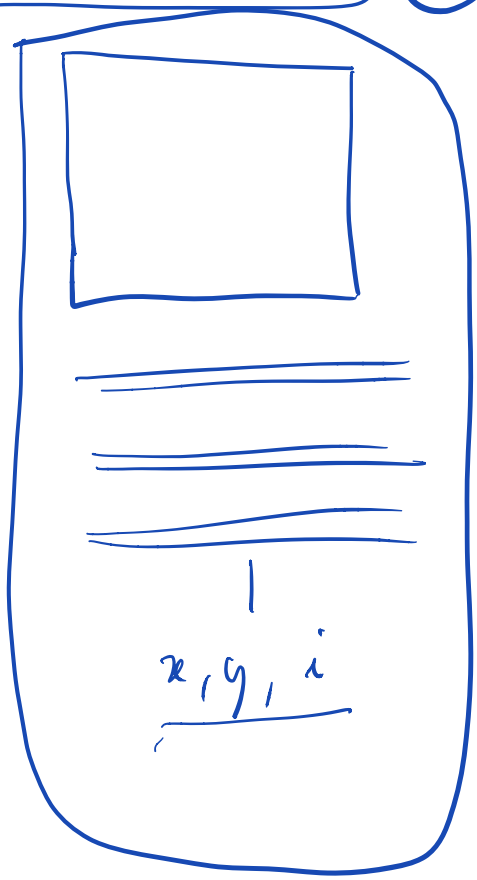


x, y, i

YOLO

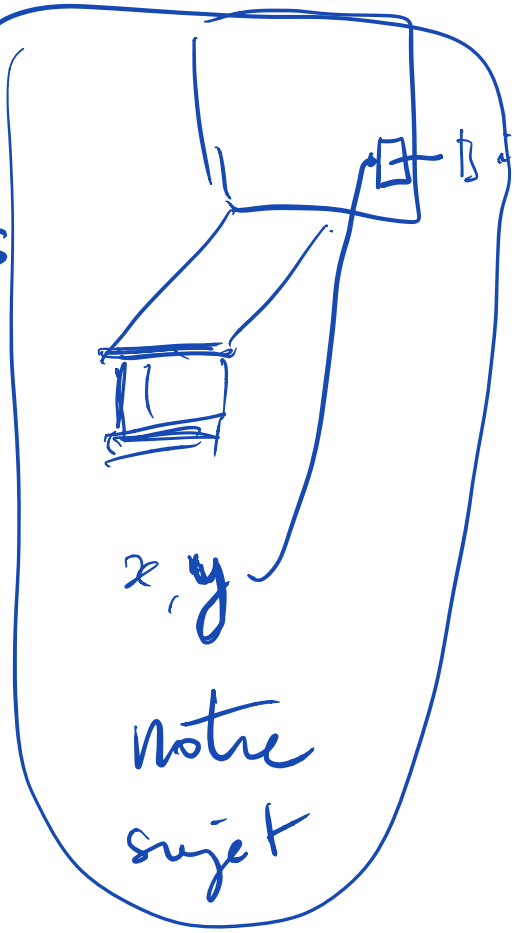
☹ BRUTE FORCE

2 hypothèses :



x, y, i

VS



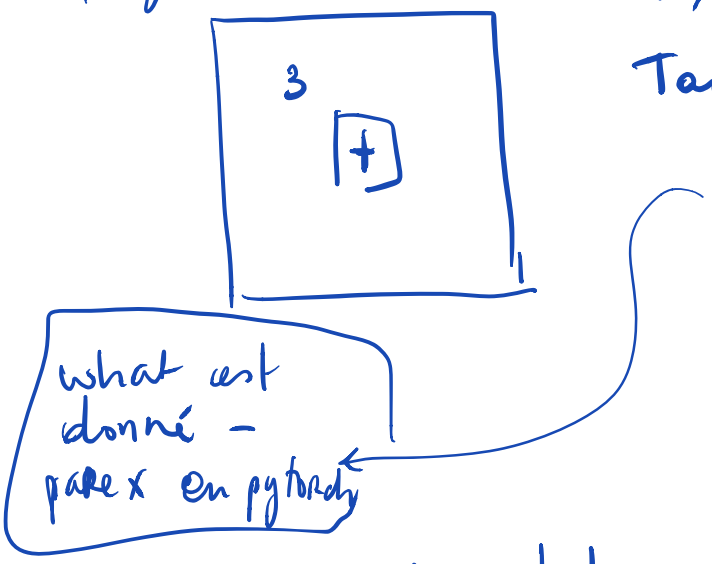
x, y

notre sujet

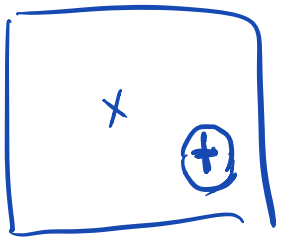
classique

(convolutionnel exhaustif)

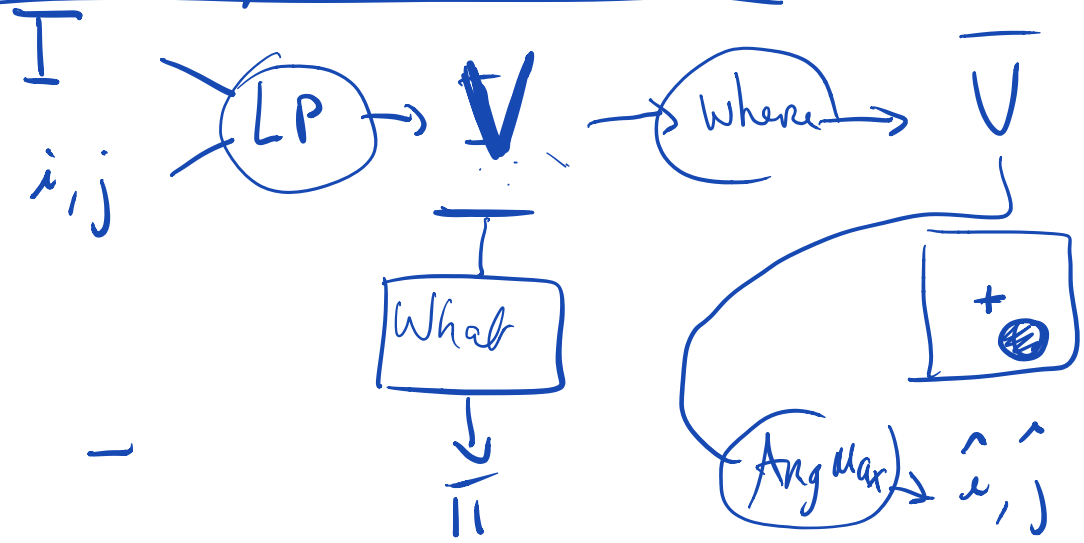
## Algorithme




## return of inhibition




## Idée computationnelle :



Peut-être résolu simplement par →

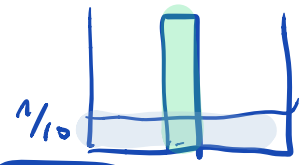
$V$ — Where → $\overline{V} = LP(\text{Kernel})$

Traduit en mots :

Sachant une image $I$ et un point de fixation $i, j$

(P) en utilisant le crop local autour du point de fixation ("macula") on peut classifier ce que l'on voit :

$$p_i = \text{What}(\text{Crop}_{i,j}(I))$$




= mauvais ⇒ on continue

= super ! ⇒ on arrête

Magno sur la périph, nous avons accès à l'information visuelle sur la carte rétinotopique

$$V = LP_{i,j}(I)$$

Nous utilisons une fonction Where qui permettent ~~d'évaluer~~ de prédire la

proba de classification en fonction de V - elle opère sur un ensemble d'hypothèse de saccades $\mathcal{H} = \{(i,j)^{(k)}\}$

(l'amplitude de la saccade est $i^{(k)} - i \,,\; j^{(k)} - j$)

$$a_i^{(k)} = \text{Where}\left(V, (i,j)^{(k)}\right)$$

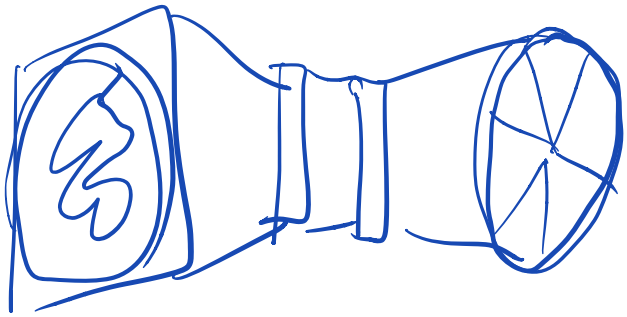
↓

proba de classification évaluée sur une base de test | sachant qu'on a effectué un mvmt vers $(i,j)^k$

Pour chaque hypothèse, (par exemple sur la grille des tous les pixels, ou mieux une grille log-polaire centrée sur l'hypothèse courante soit $(i,j)$ ).

Le but est d'apprendre la fonction qui génère les $a_i^{(k)}$ sur les points $(i,j)^{(k)}$ pour un V donné

→ c'est de l'apprentissage supervisé :

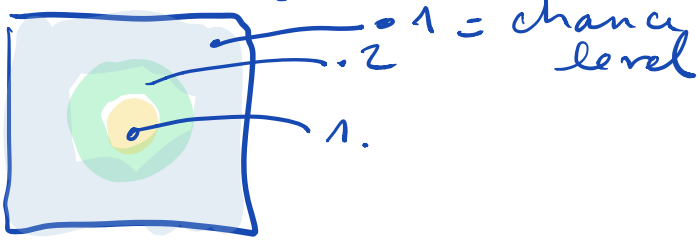


V a

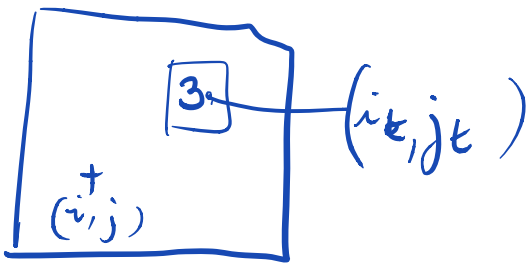
⇒ on peut l'apprendre par une descente de gradient en back prop

• Computationnellement, il peut paraître fastidieux de faire toutes les hypothèses à chaque saccade (même si c'est ce qu'on fait de notre vivant...) et on peut utiliser le kernel de 2018-02-16_classifier.ipynb :

- On connait pour une erreur de placement $\Delta i, \Delta j$ la carte des $a_i$

$A(\Delta i, \Delta j) =$ 

- Sur une image où on connaît la vraie position de l'objet $(i_t, j_t)$




on a pas besoin de calculer $a_i$ sur tous les points car ça dépend pas de $(i, j)$ mais seulement de

$$\Delta i, \Delta j = i^{(k)} - i \;,\; j^{(k)} - j$$

et donc pour chaque hypothèse,

$$a^{(k)} = A\left(i^{(k)} - i \;,\; j^{(k)} - j\right)$$

en particulier, pour une liste de points H sur la log-polaire, ça ressemble à une transfo log-polaire

## Priorités

- collecter des paires $(V, \star)$
- apprendre à prédire $\star$ sachant $V$
- appliquer à un cas plus complexe avec du bruit.