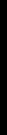


# Contrôle d'un bras robotique sous-marin en apprentissage par renforcement

Soutenance de trimestre recherche  
10 février 2023



XU Xindong  
Cycle ingénieur civil, 2ème année  
Ecole de Mines de Paris, PSL



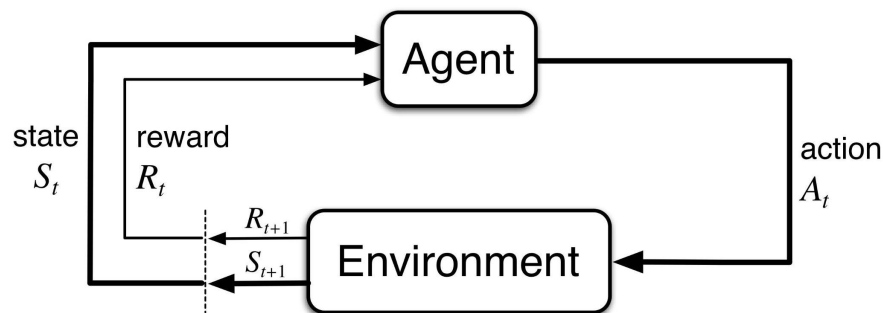
# 1 Contexte de la recherche - Robots sous-marin



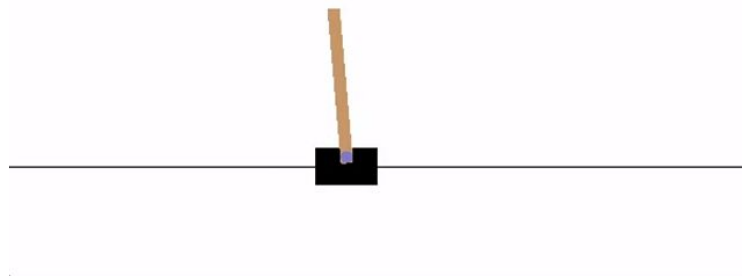
Difficultés à surmonter : pression élevée, visibilité réduite, loi de commande compliquée, robot souple, low-cost, reaching task à un point donné...

L'objectif principal : développer un contrôleur pour un bras robotique sous-marin qui va **guider l'extrémité du bras vers une position indiquée** par l'opérateur en utilisant des techniques d'apprentissage par renforcement pour surmonter ces défis complexes.

# 1 Contexte de la recherche - Reinforcement Learning

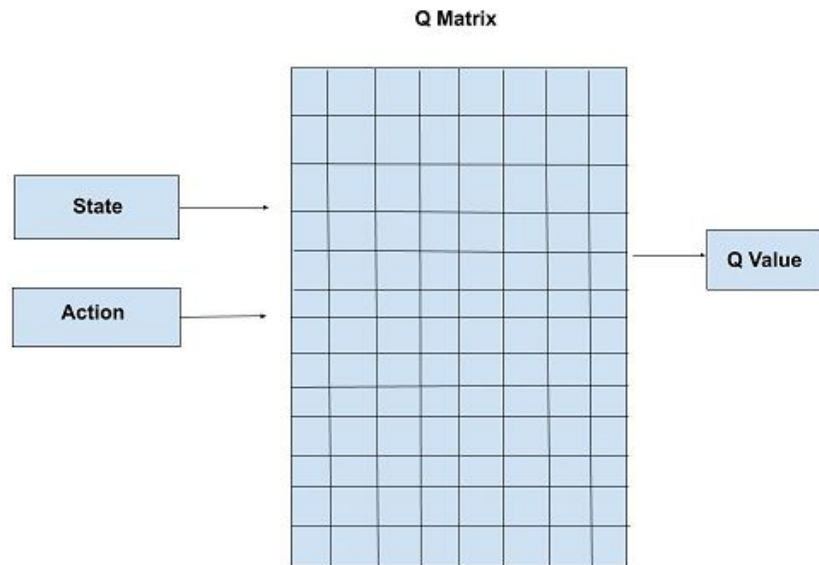


- Observation de son environnement (state)
- Décision en fonction de cette observation (action)
- Rétroaction sous forme de récompense (reward)
- Update de stratégie pour maximiser la récompense à long terme



## 2 Méthodologie - Q Learning

Q value table	a1	a2
s1	q(s1, a1)	q(s1, a2)
s2	q(s2, a1)	q(s2, a2)
s3	q(s3, a1)	q(s3, a2)



$$Q(s, a) \leftarrow Q(s, a) + \alpha(R(s, a) + \gamma \max_a Q(S', a) - Q(s, a))$$

$\alpha$  est le taux d'apprentissage

$\gamma$  est le facteur de réduction de l'horizon temporel

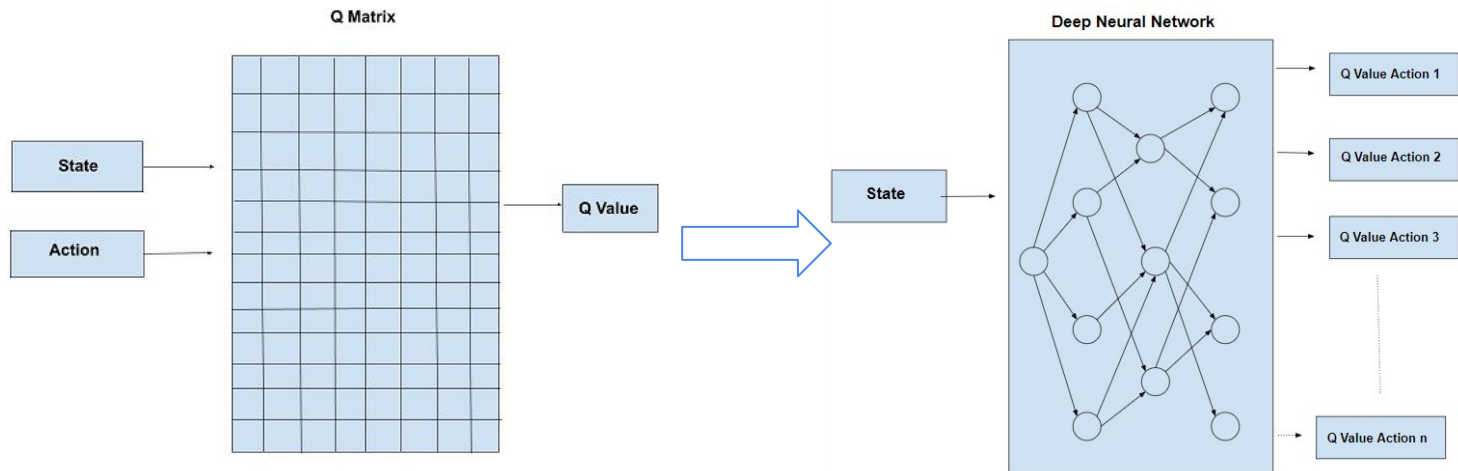
$R$  est la récompense immédiate reçue par l'agent en prenant l'action  $a$  dans l'état  $s$

$s'$  est l'état suivant dans lequel l'agent se trouve après avoir pris l'action  $a$

$a'$  est l'action choisie dans l'état  $s'$

## 2 Méthodologie - Deep Q Learning

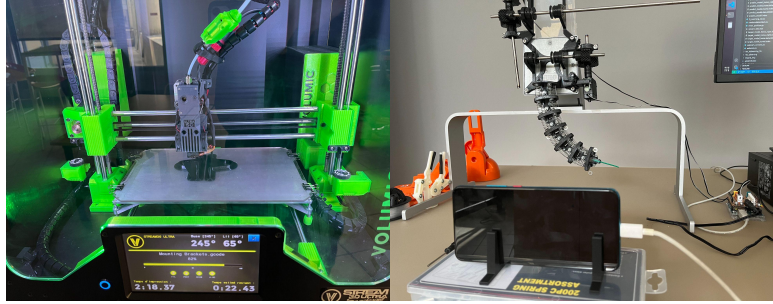
$$\theta_{k+1} = \theta_k - \alpha \Delta_{\theta} E_{s' \sim P(s'|s,a)} [(Q_{\theta}(s,a) - \text{target}(s'))^2] |_{\theta=\theta_k}$$



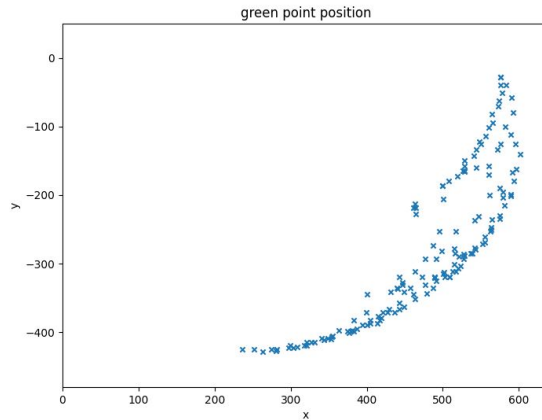
Il présente plusieurs avantages par rapport au Q-Learning traditionnel :

1. Traitement des états continus
2. Apprentissage en temps réel : Le Deep Q-Learning peut effectuer un apprentissage en temps réel en utilisant des algorithmes d'optimisation tels que le gradient descendant avec rétropropagation
3. Traitement de grands ensembles de données
4. Amélioration de la stabilité et de la rapidité d'apprentissage

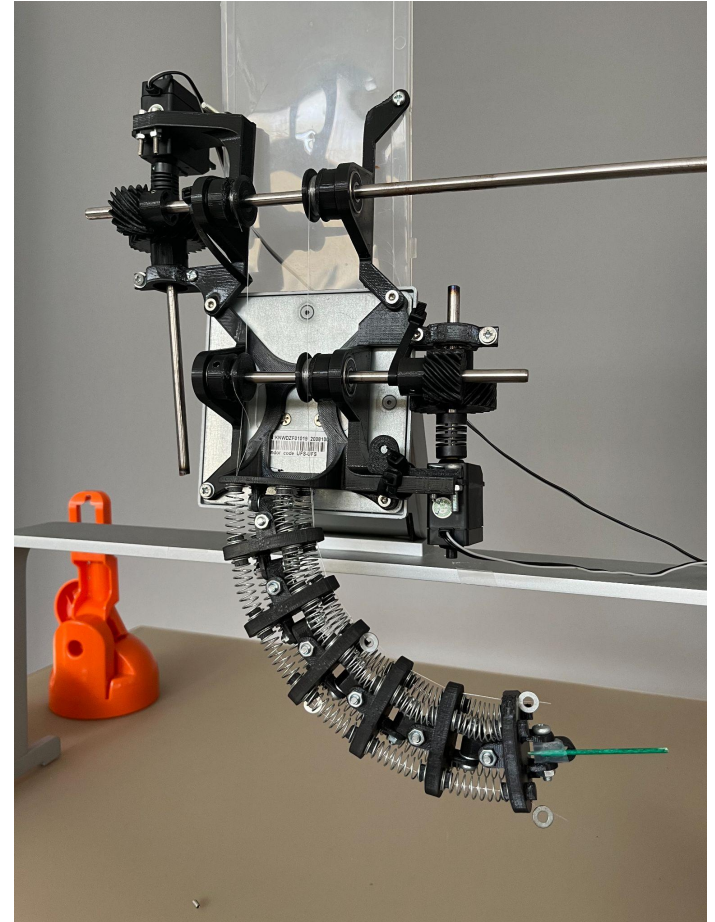
## 2 Méthodologie Expérimentale



pièces de robots imprimées en 3D capture de position du bras avec un caméra



positions potentiellement atteignables du sommet



## 2 Méthodologie Expérimentale - Network and training design

$Q(s, a | \theta)$ ,

avec

$s$  = (position actuelle, position cible)

$a$  = (input de moteur1, input de moteur2)

$R$ , la distance euclidienne entre les deux positions

Model: "sequential"

Layer (type)	Output Shape	Param #
batch_normalization (Batch Normalization)	(None, 6, 1)	4
dense (Dense)	(None, 6, 6)	12
batch_normalization_1 (Batch Normalization)	(None, 6, 6)	24
dense_1 (Dense)	(None, 6, 4)	28
batch_normalization_2 (Batch Normalization)	(None, 6, 4)	16
dense_2 (Dense)	(None, 6, 1)	5

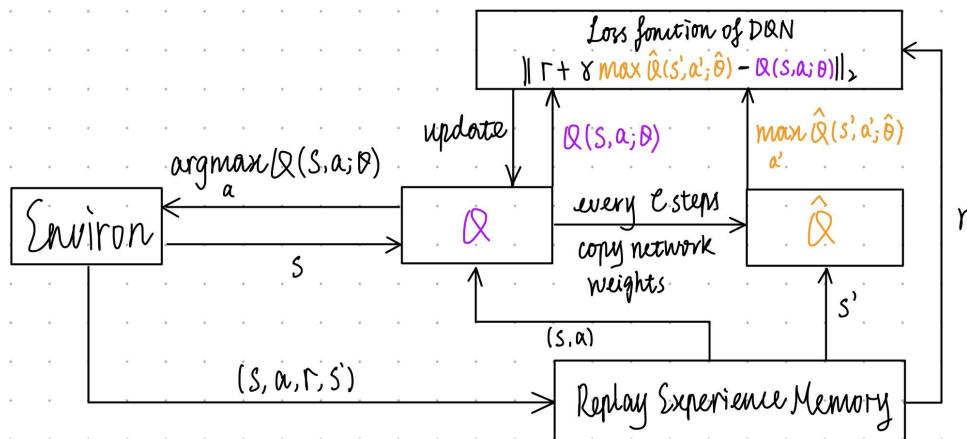
Total params: 89

Trainable params: 67

Non-trainable params: 22



## 2 Méthodologie - Network and training design



### Algorithm 1: deep Q-learning with experience replay.

Initialize replay memory  $D$  to capacity  $N$

Initialize action-value function  $Q$  with random weights  $\theta$

Initialize target action-value function  $\hat{Q}$  with weights  $\theta^- = \theta$

**For** episode = 1,  $M$  **do**

Initialize sequence  $s_1 = \{x_1\}$  and preprocessed sequence  $\phi_1 = \phi(s_1)$

**For**  $t = 1, T$  **do**

With probability  $\epsilon$  select a random action  $a_t$

otherwise select  $a_t = \operatorname{argmax}_a Q(\phi(s_t), a; \theta)$

Execute action  $a_t$  in emulator and observe reward  $r_t$  and image  $x_{t+1}$

Set  $s_{t+1} = s_t, a_t, x_{t+1}$  and preprocess  $\phi_{t+1} = \phi(s_{t+1})$

Store transition  $(\phi_t, a_t, r_t, \phi_{t+1})$  in  $D$

Sample random minibatch of transitions  $(\phi_j, a_j, r_j, \phi_{j+1})$  from  $D$

Set  $y_j = \begin{cases} r_j & \text{if episode terminates at step } j+1 \\ r_j + \gamma \max_{a'} \hat{Q}(\phi_{j+1}, a'; \theta^-) & \text{otherwise} \end{cases}$

Perform a gradient descent step on  $(y_j - Q(\phi_j, a_j; \theta))^2$  with respect to the network parameters  $\theta$

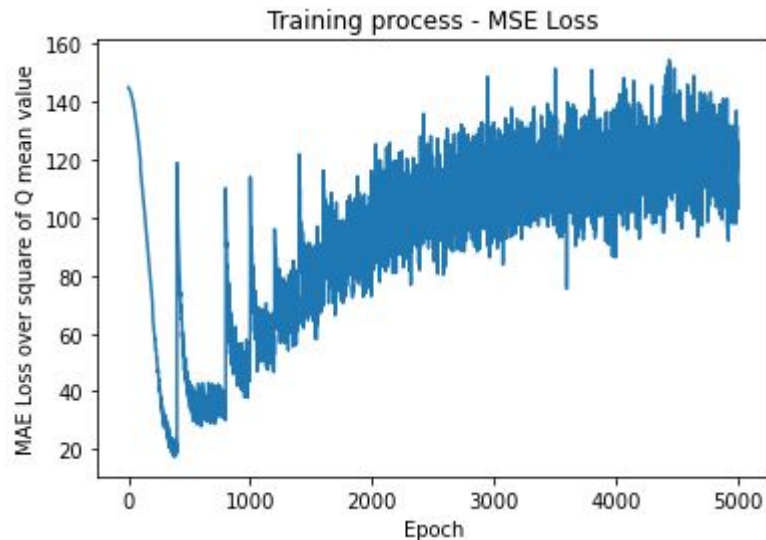
Every  $C$  steps reset  $\hat{Q} = Q$

**End For**

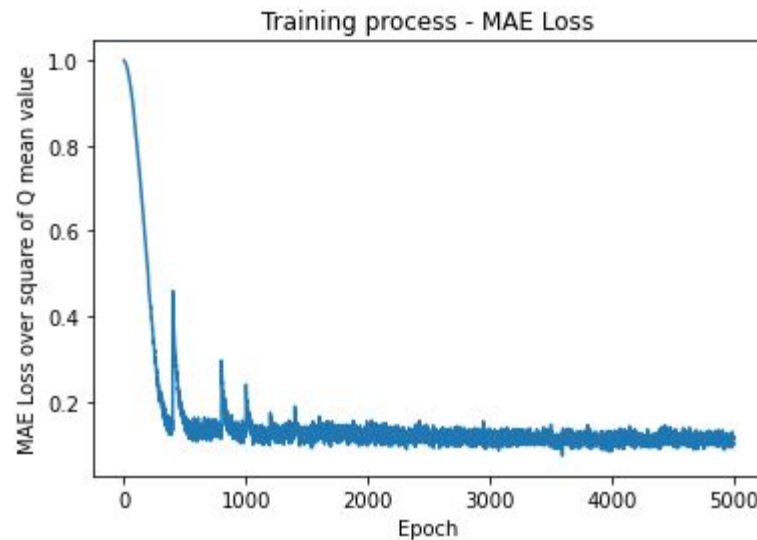
**End For**



### 3 Résultat - Training process



évolution de la fonction objectif avec epoch de training



évolution de la fonction objectif sur la moyenne de Q value avec epoch de training

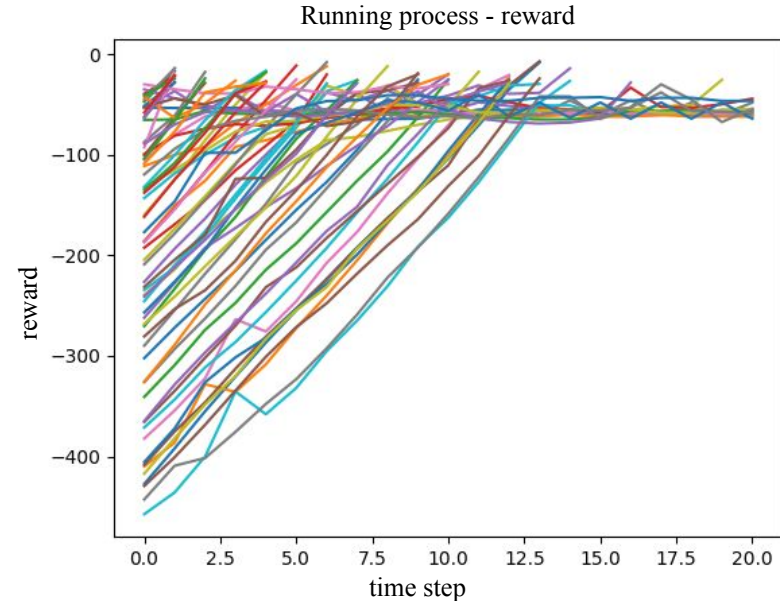
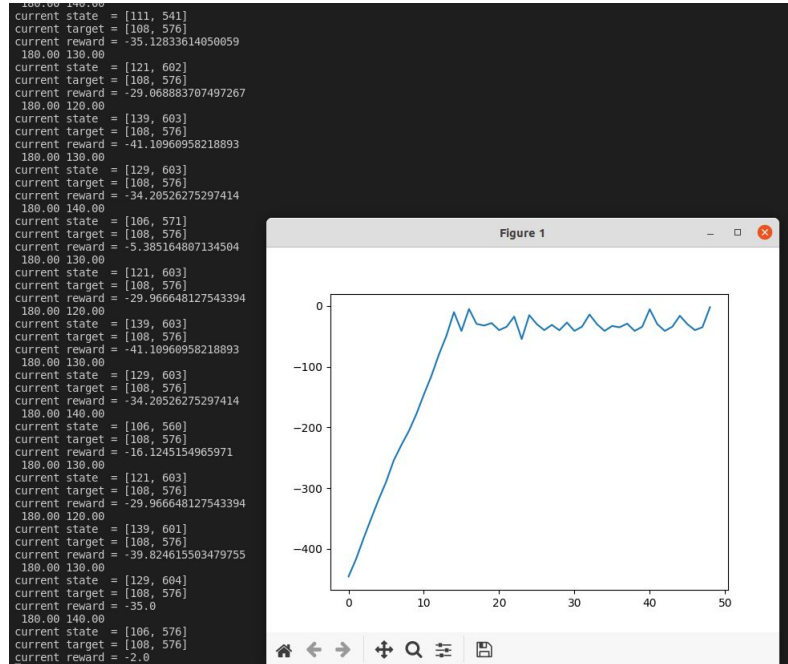
### 3 Résultat - Implémentation de l'algorithme

```
_model_home.night.index  
_model_home.data-00000-of-00001  
_model_home.index  
_model_jeuudi.data-00000-of-... M WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).opt  
_model_jeuudi.index M WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).opt  
_model_monday.data-00000-of-0... WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).opt  
_model_monday.h5 WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).opt  
_model_monday.index WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).opt  
_model_ok.data-00000-of-00001 WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).opt  
_model_ok.index WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).opt  
_model_testtest.data-00000-of-0... WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).opt  
_model_testtest.index WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).opt  
_model_weekend.data-00000-of-... WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).opt  
_model_weekend.index WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).opt  
_model_.data-00000-of-00001 WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).opt  
_model_.index WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).opt
```

mig5@mig5-Precision-T1700: ~/Desktop/TR\_DATA\_RL\$ /usr/bin/python3 /home/mig5/Desktop/TR\_DATA\_RL

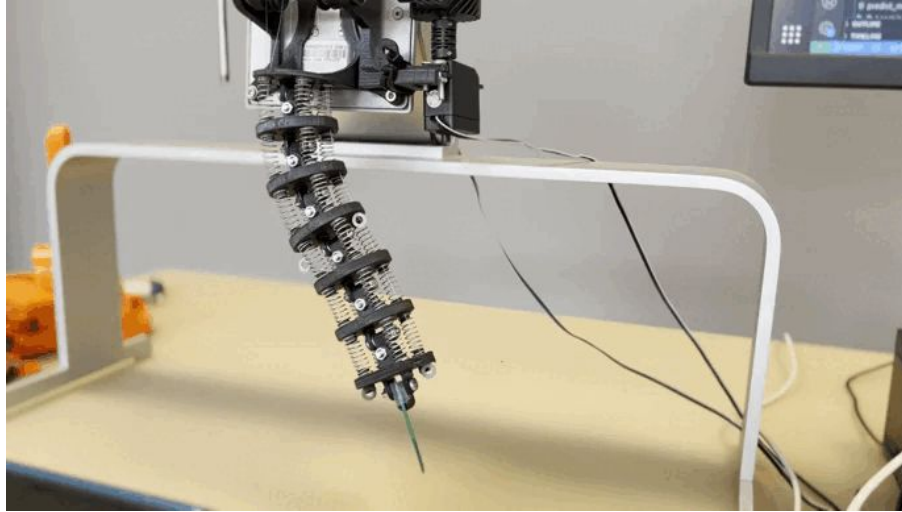
138.00 160.00

# 4 Discussion - Conclusion



Si on définit une récompense finale de moins de 15 pixels comme une réussite de la tâche, alors notre modèle actuel peut atteindre une précision de 82,8%.

# 4 Discussion - Conclusion



On a développé un contrôleur en 2D basé sur deep q learning de l'apprentissage par renforcement pour un bras robotique souple qui va guider l'extrémité du bras vers une position indiquée par l'opérateur au début de chaque essaie.

Notre modèle actuel peut atteindre une précision de 82,8%.

# 4 Discussion - Limites et problèmes de la recherche

Pour l'instant, notre recherche rencontre encore les problèmes et les limites suivants :

Détection imprécise de la position

Contrôle du moteur imprécis, structure de mécanique instable

Contrôle du bras de mouvement en 3D

# Merci pour votre attention