



SCHWEIZER JUGEND FORSCHT
SCIENCE ET JEUNESSE
SCIENZA E GIOVENTÙ

Mathématiques/Informatique



Mathéo Beney, 30.09.2003



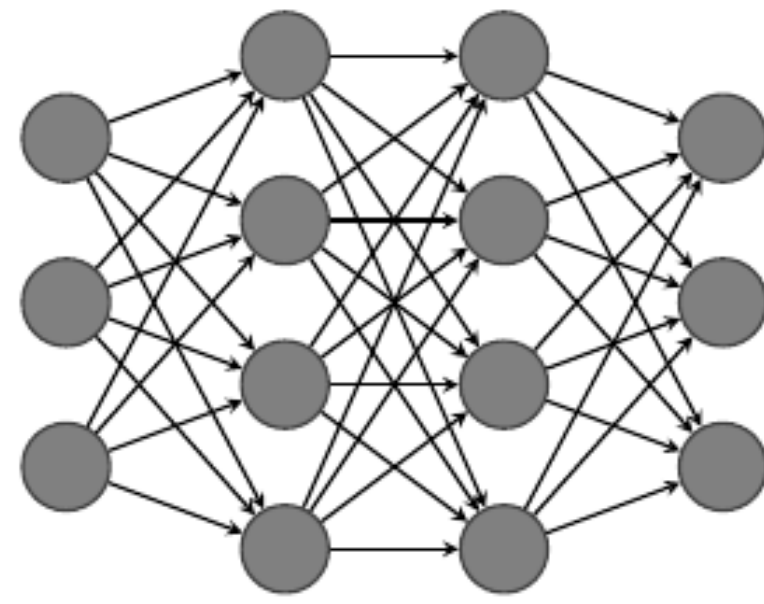
Initiation à l'intelligence artificielle au travers de projets

Problématique

L'objectif de ce travail est d'utiliser différents projets en intelligence artificielle pour comprendre son fonctionnement et ses limites dans une démarche d'apprentissage autodidacte.

Méthode

À partir des éléments théoriques trouvés dans la littérature, nous avons utilisé le langage de programmation python et la librairie Pytorch pour élaborer les différents projets présentés ci-dessous, c'est-à-dire la création de modèles, leur utilisation et leur entraînement.



Représentation schématique d'un réseau de neurones.

$$\begin{matrix} \text{Entrée} & & \text{Poids} & & \text{Biais} \\ \begin{bmatrix} i_1 & i_2 & i_3 \end{bmatrix} \cdot \begin{bmatrix} w_{1,1} & w_{2,1} & w_{3,1} & w_{4,1} \\ w_{1,2} & w_{2,2} & w_{3,2} & w_{4,2} \\ w_{1,3} & w_{2,3} & w_{3,3} & w_{4,3} \end{bmatrix} & + & \begin{bmatrix} b_1 & b_2 & b_3 & b_4 \end{bmatrix} \end{matrix}$$

Calculs mathématiques d'un réseau à l'aide de matrices.

```
for epoch in range(10, EPOCHS):
    for i, data in enumerate(data_loader):
        model.train()
        x, y_true = data[0], data[1]
        optimizer.zero_grad()
        y = model(x)
        loss = criterion(y, y_true)
        loss.backward()
        optimizer.step()
    model.eval()
    total = 0
    correct = 0
    for testdata in dataloader:
        xtest, y_true_test = testdata[0], testdata[1]
        ytest = model(xtest)
        prediction = model.predictions(ytest)
        total += y_true_test.size(0)
        correct += (prediction == y_true_test).sum().item()
    acc = 100 * correct / total
    print(f'validation acc: {acc} (total: {total}) testimages: {acc}')
```

Développement en python d'un réseau de neurones.

Résultats

Snake et Flappy Bird

Modèle

- Réseaux linéaires

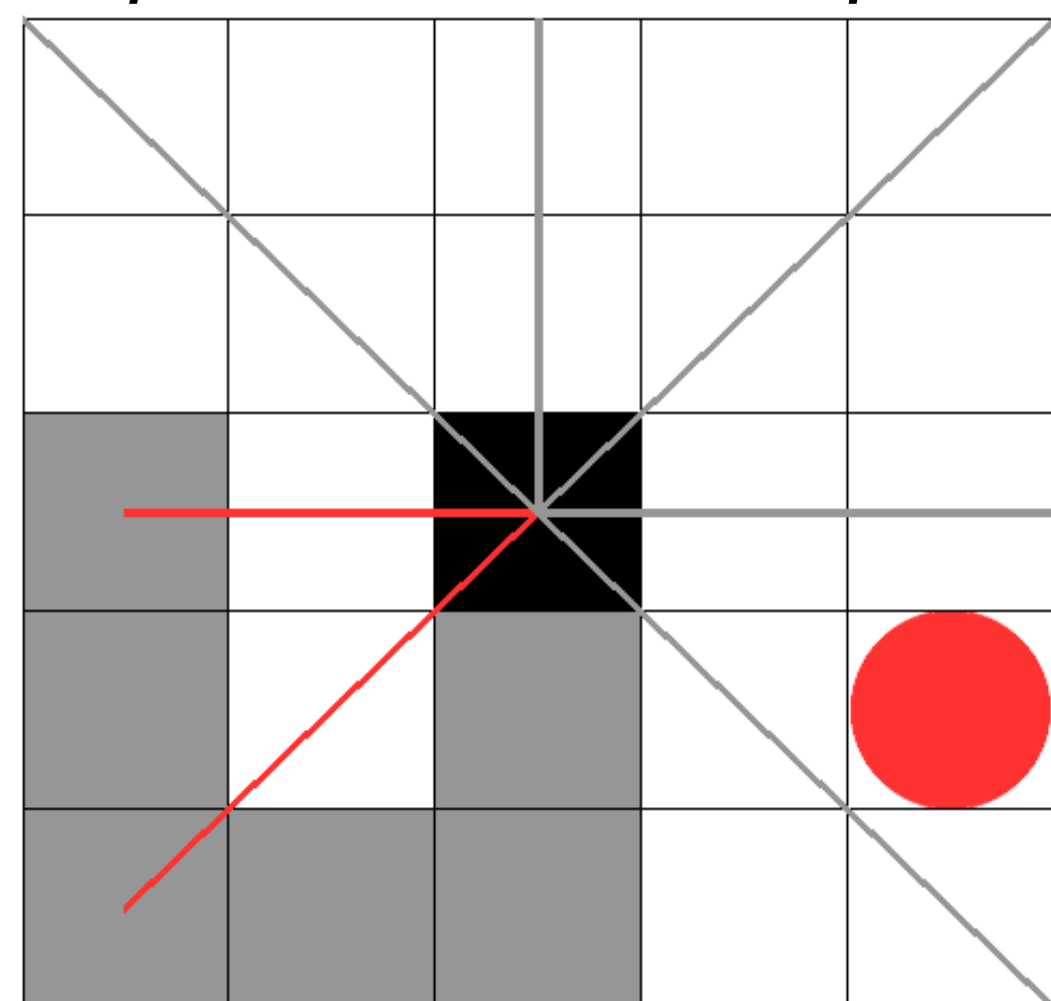
Dataset

- État du jeu + récompense

Spécificités

- Algorithme d'évolution
- Deep Q-learning

Représentation des inputs

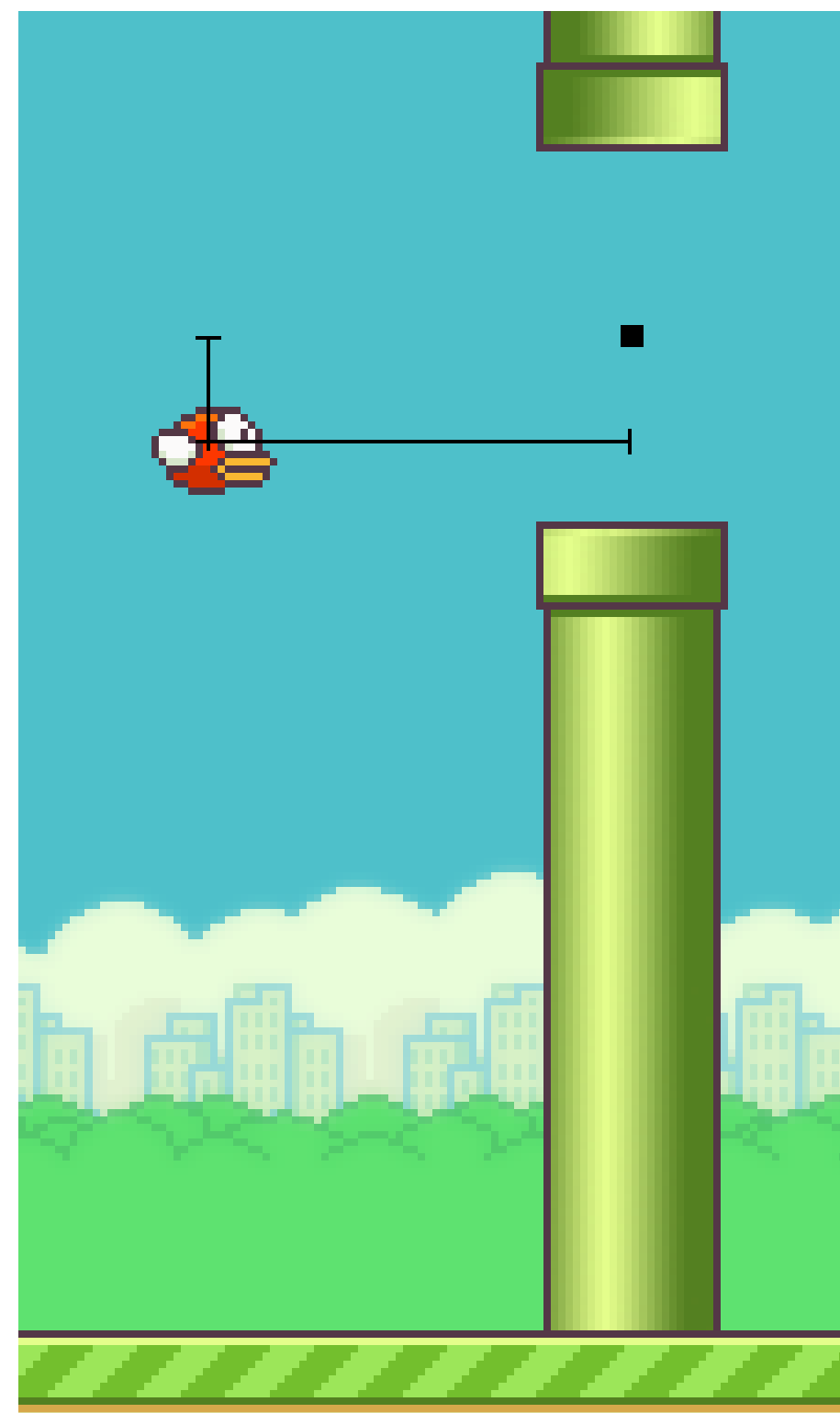


Snake

Algorithme d'évolution

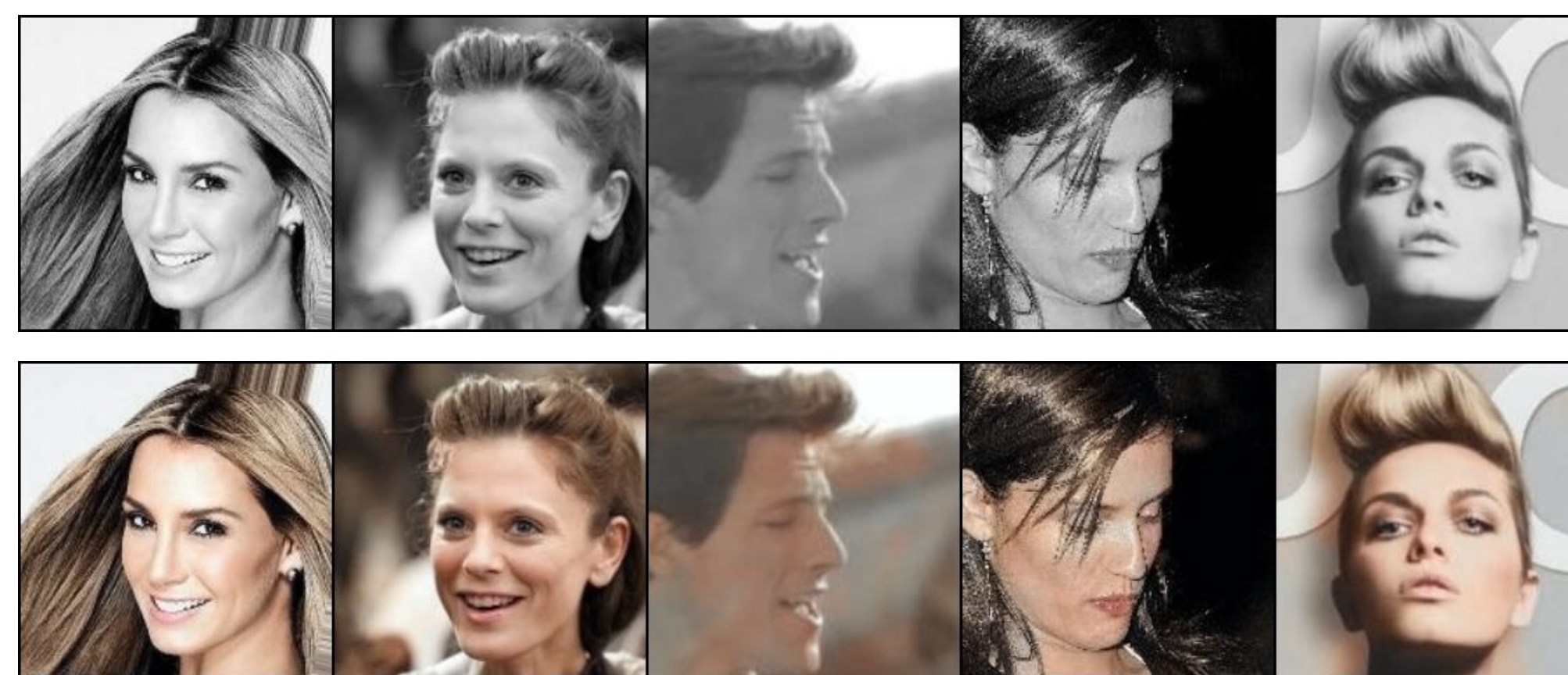
Deep Q-learning

Flappy Bird



Algorithme d'évolution

Colorisation



Modèle

- Couches de convolution
- Transfer learning (*Resnet*)

Dataset

- CelebA (200'000 photos)

Spécificité

- Utilisation du format Lab pour les couleurs

Segmentation



Modèle

- Couches de convolution
- Transfer learning (*Resnet*)

Dataset

- Créé par Supervisely

Spécificité

- Data augmentation

Transfert de style

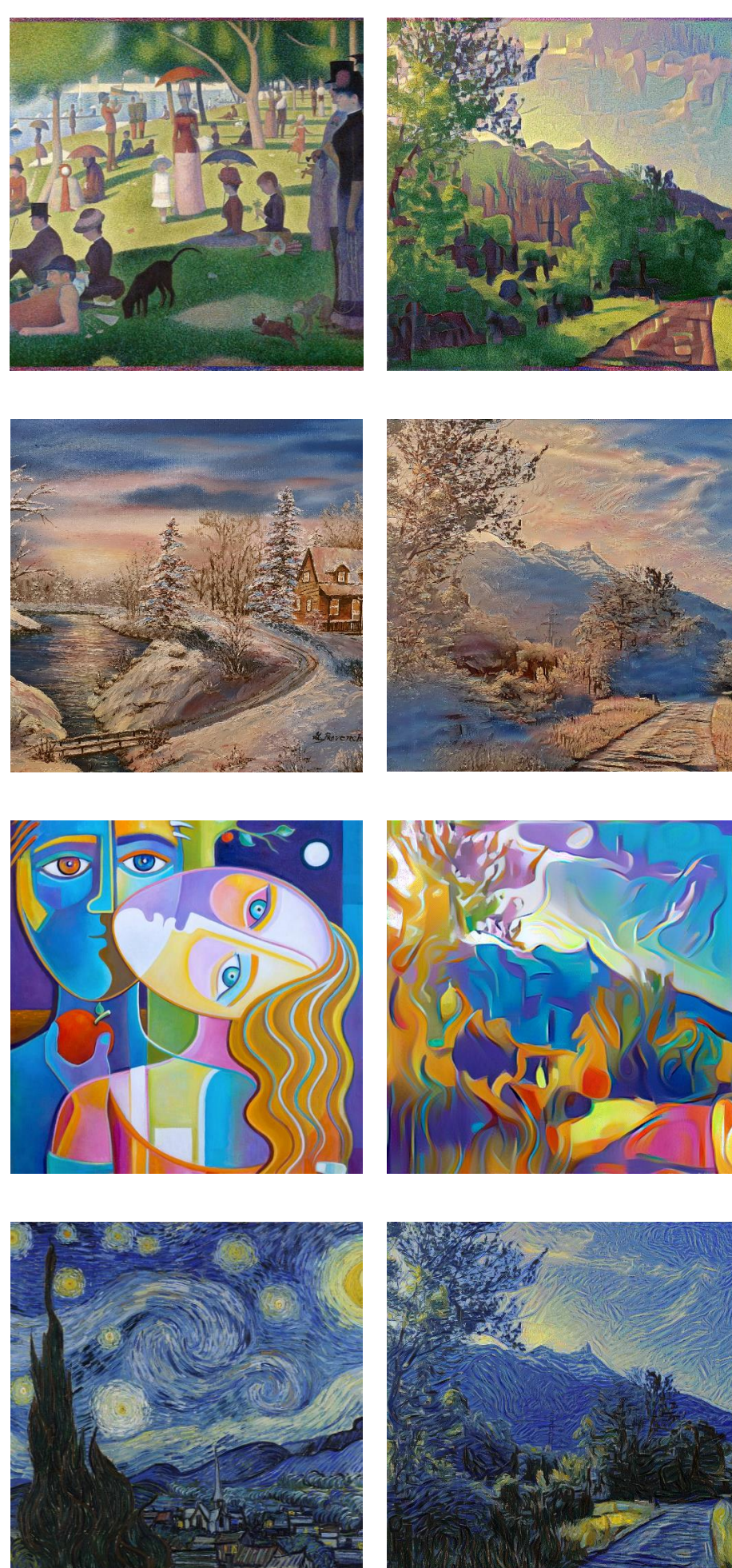


Modèle

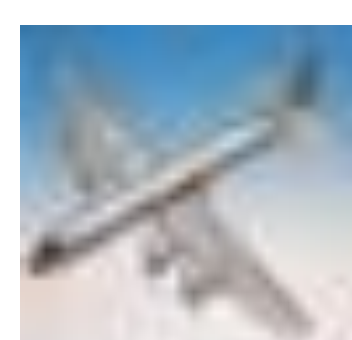
VGG

Spécificité

- Entraînement non nécessaire
- Rétropropagation sur les entrées



Classification



modèle

Prédiction de classe

10 classes

• 87.59%

100 classes

• Moyen: 67.92%

• Transfer learning: 77.26%

Modèle

- Couches de convolution

Dataset

- CIFAR10 et CIFAR100

Spécificité

- Plusieurs tailles de modèle

Génération de visages



Modèle

- 2 réseaux:
 - discriminateur (convolution)
 - générateur (convolution transposée)

Dataset

- CelebA (200'000 photos)

Spécificité

- 2 essais → RGB & tons de gris

Discussion

Plusieurs pistes d'améliorations nous semblent possibles (taille du réseau, temps d'entraînement). Nous n'avons ni le temps, ni les équipements nécessaires pour tester ces pistes. Bien que nous ne l'ayons qu'effleuré, nous avons pu expérimenter les difficultés et les potentialités de l'IA à partir de notre démarche autodidacte.

Conclusion

L'objectif principal de ce travail a été atteint, car il nous a permis de nous confronter concrètement au monde de l'IA. Sa mise en pratique au travers des projets, nous a permis de remettre en cause des préjugés, d'approfondir des aspects théoriques et d'identifier nos lacunes.