

# CRAPAUDUC

Loïc Haas – Carol Hubert  
Machine Learning on Big Data  
Juin 2018 – Master MSE

## 1. Introduction

Ce projet se base sur des photos récoltées dans le crapauduc de Gimel, où la prise d'image est déclenchée par les mouvements, lors du passage d'un animal bien sûr, mais aussi lorsque les feuilles sont déplacées par le vent par exemple. Les changements brutaux de luminosité, dus au passage d'une personne dans la zone de l'objectif, sont également des éléments déclencheurs des caméras. Une photo est alors enregistrée chaque seconde, et notre intention avec ce travail était de déterminer pour chaque image si elle ne contient pas d'animal, au moins un triton ou au moins une "*crapouille*" (crapaud ou grenouille, trop difficiles à différencier l'un de l'autre sans être un spécialiste). Pour ce faire, nous nous sommes basés sur des photos déjà labellisées par des êtres humains, l'objectif final du projet étant de compter le nombre de batraciens qui utilisent effectivement le crapauduc. Celui-ci est composé de 18 canaux souterrains ; chacun de ces tunnels est équipé d'un appareil photographique dont l'objectif est centré sur une planche transversale au canal.

## 2. Base de données

La base de données complète contient environ 960'000 images non labellisées, pour un volume total de plus de 500 Gigabytes de données. Ces photos sont prises aussi bien de jour, en couleurs, que de nuit, en nuances de gris. Leur résolution est de 1920 x 1080. Seules 13'000 de ces images ont été étiquetées jusqu'ici, dont plus de 10'000 ne contiennent aucun batracien. La majorité des images labellisées font partie de *séquences* continues. Une séquence est une rafale d'images, espacées d'une seconde, prises après le déclenchement de l'appareil par un mouvement jusqu'à ce que celui-ci s'arrête. Si les animaux photographiés ne se trouvent pas forcément sur la planche située à l'entrée du canal, ils y passent tous à un moment ou à un autre. Les photos prises ne sont pas réparties uniformément dans tout le crapauduc, car les canaux centraux connaissent un trafic plus élevé que ceux des bords.

Nous disposons également, pour les séquences de photos labellisées, de statistiques recueillies par l'assistant responsable du projet, monsieur Hector Satizabal. Ces données sont les suivantes : *camera*, *start*, *end*, *max*, *mean*, *n*, soit le numéro de l'appareil photo (du canal correspondant), le timestamp de début de la séquence, celui de la fin, la différence maximale entre deux images de la séquence, la différence moyenne, et le nombre d'images contenu dans la séquence.

### 3. Prétraitement des données

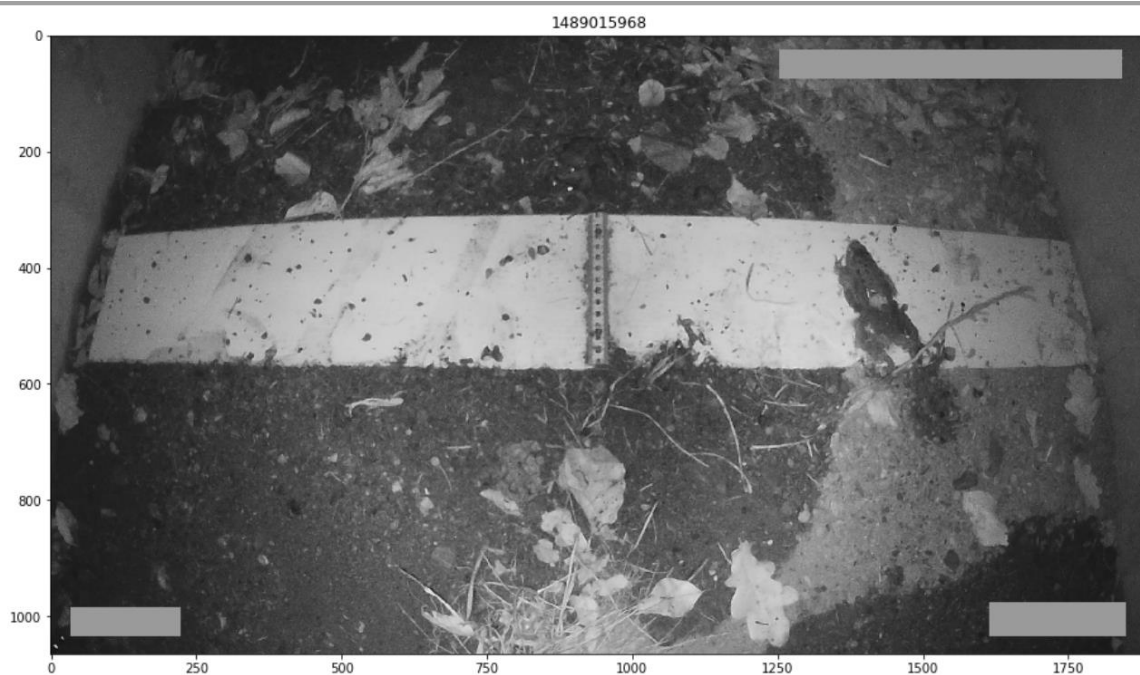
Nous avons travaillé sur les photos déjà labellisées, ce qui nous permettait d'avoir une référence pour l'évaluation des résultats. Nous avons commencé par regrouper toutes les séquences comprenant au moins une photo étiquetée *toad* (crapaud), *frog* (grenouille) ou *newt* (triton).

Partant de l'idée que l'objectif final est de compter le nombre d'animaux traversant le crapauduc, et pas de labelliser chaque image proprement dite, nous avons choisi d'analyser le mouvement, défini par les différences entre deux prises de vue. En effet, pour chaque mouvement, un animal a une position d'arrivée différente de sa position de départ. Et pour chaque animal, il y aura forcément du mouvement entre le moment où il entre dans la zone couverte par la caméra et le moment où il en sort, même si entre deux il reste immobile pendant un temps plus ou moins long. Typiquement, si les tritons avancent régulièrement mais pas très vite, une *crapouille* se déplace plutôt par saut, avec des pauses immobiles entre deux, couvrant ainsi une plus grande distance pour chaque mouvement, mais pas forcément plus rapidement.

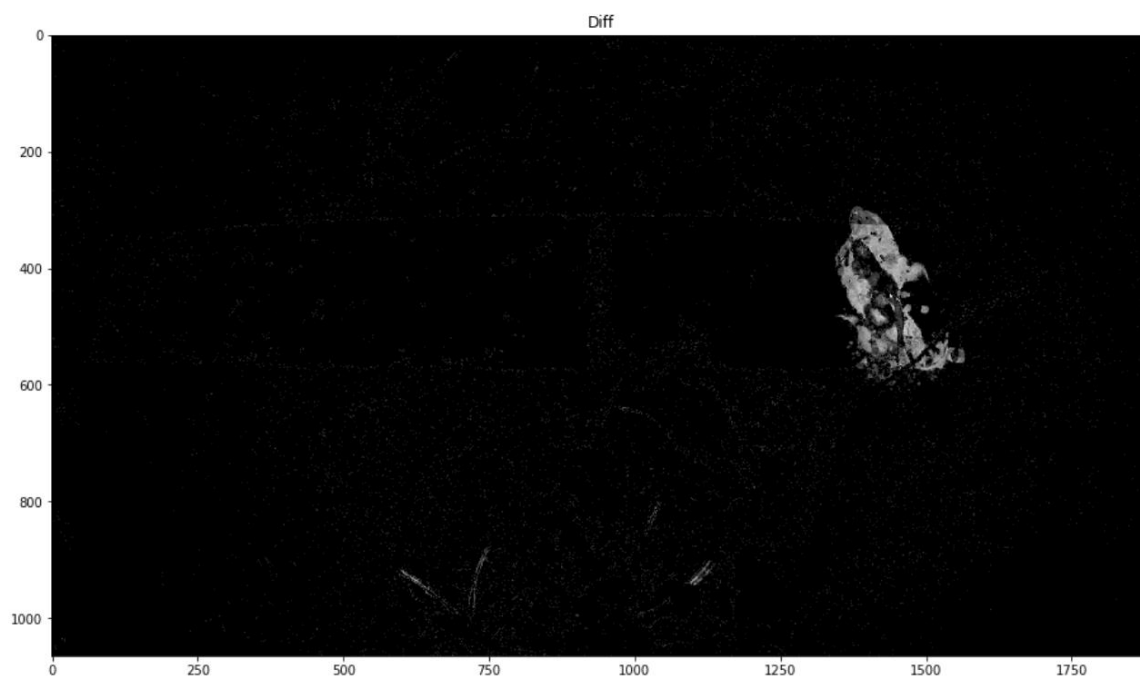
Dans un premier temps, pour chaque séquence traitée, nous avons calculé la différence entre chaque image et la suivante. Nous avons pour cela utilisé la bibliothèque openCV, plus particulièrement la fonction `diff = cv2.absdiff(previous, img)`. Par la suite, toute différence ainsi obtenue entre deux pixels mais inférieure à 15 a été éliminée car trop insignifiante – on considère qu'elle est inexistante. Pour affiner encore le résultat, nous avons appliqué la méthode de traitement d'image appelée érosion (*erode* – réduction du bruit) disponible dans cette bibliothèque, puis pour réunir les éléments disjoints mais relativement proches, nous avons pratiqué la dilatation (*dilate*).

Chaque élément ainsi isolé est ensuite affiché séparément, voire agrandi, et son contour tracé sur l'image initiale. L'image d'arrivée devient enfin à son tour une image de départ, et ainsi de suite jusqu'à la fin de la séquence.

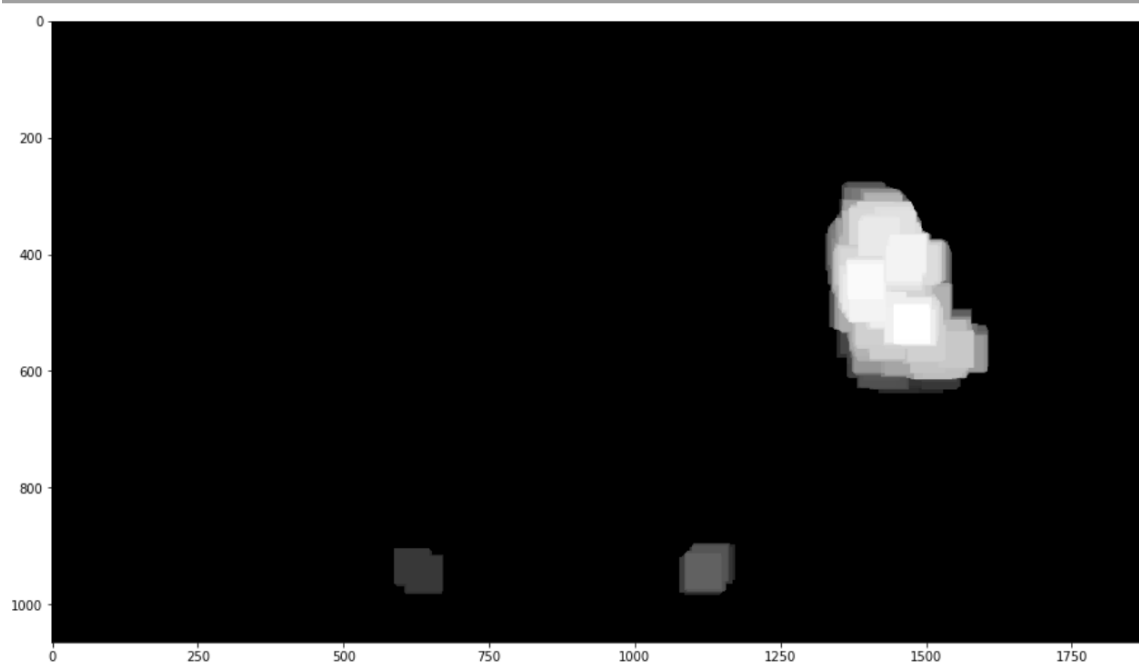
Ci-après un exemple des résultats obtenus pour deux images :



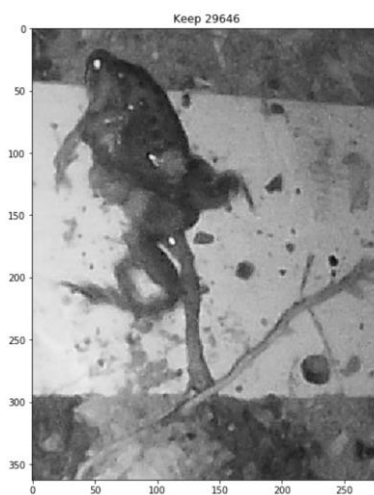
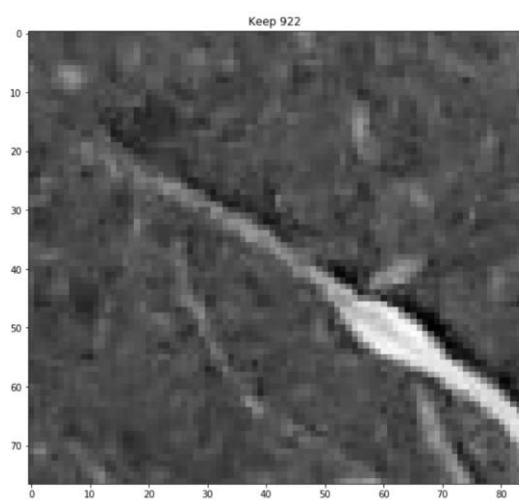
*Image de départ*



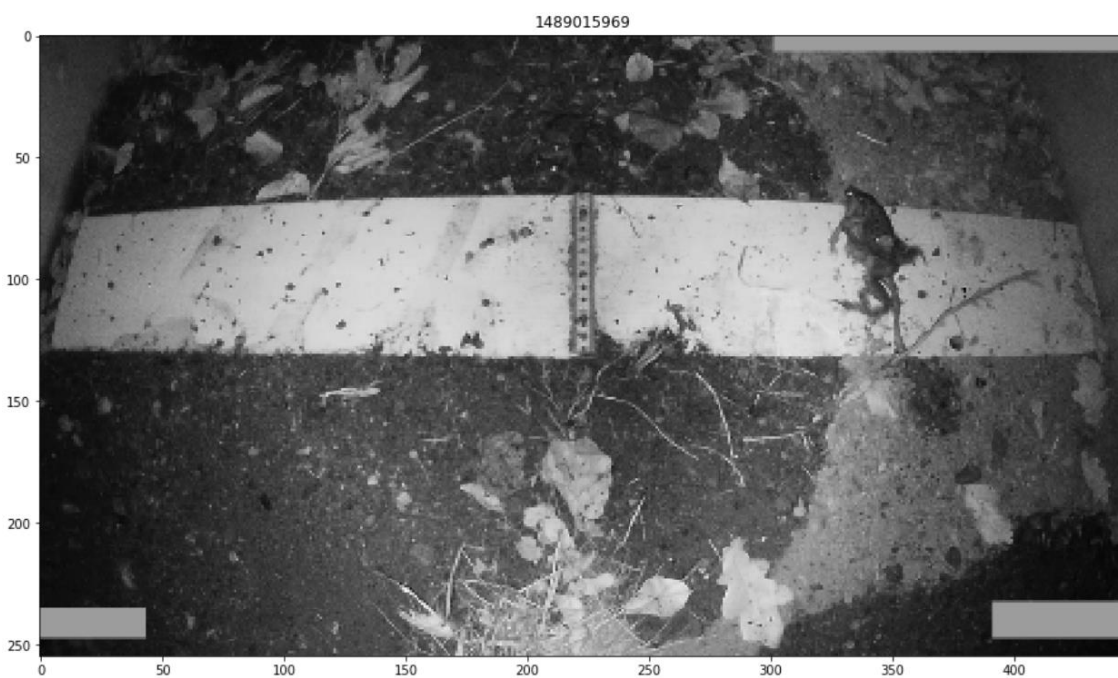
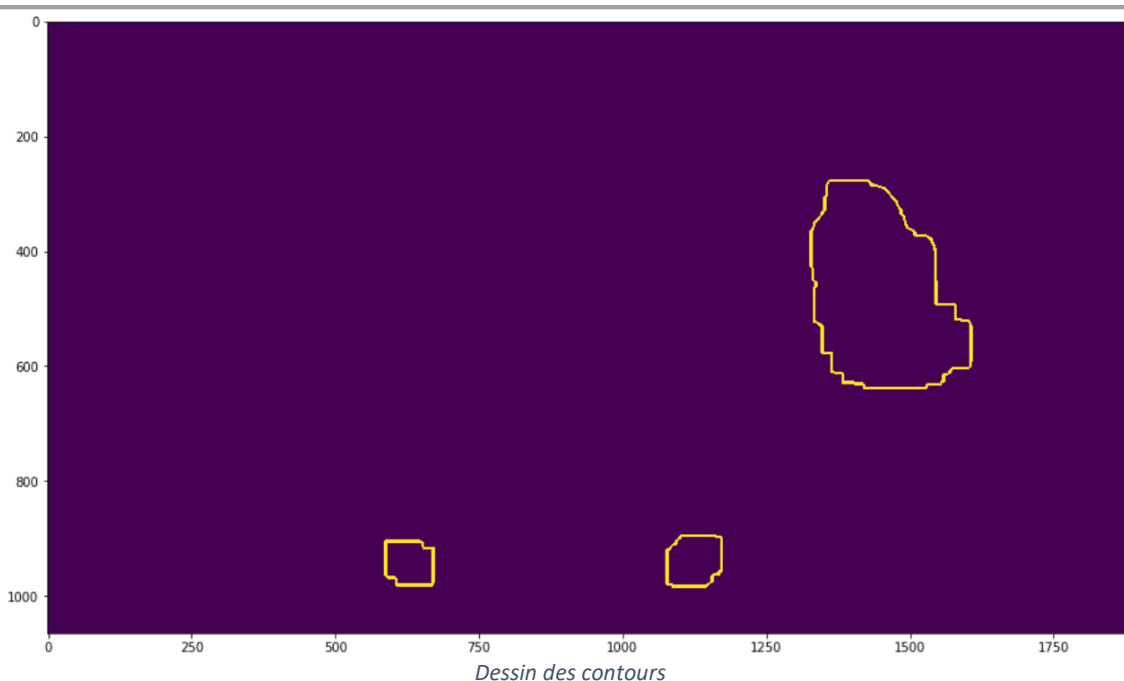
*Différence avec l'image d'arrivée*



*Différence érodée, puis dilatée*



*Les trois éléments isolés*

*Image d'arrivée*

Ayant choisi d'utiliser, pour le Machine Learning, un réseau de neurones à convolution pré-entraîné (VGG16) qui prend en entrée des images d'une résolution fixe de 224x224 pixels, notre pré-traitement d'image comprend également un redimensionnement des éléments extraits : largeur et hauteur du rectangle circonscrit sont comparées, et la plus petite dimension est ramenée à la plus grande, évitant ainsi une perte d'information. Ce carré est enfin ajusté à 224x224.

## 4. Machine Learning

Pour l'analyse des animaux sur les images, nous avons choisi, comme mentionné auparavant, d'utiliser un réseau de neurones à convolution (CNN), puisqu'il s'agit là d'un outil idéal pour la reconnaissance d'image. Plutôt que de partir de zéro, nous avons préféré prendre un CNN pré-entraîné avec un millier d'objets ou animaux divers, VGG-16.

Les réseaux de neurones pré-entraînés avec énormément d'images, comme VGG-16 ou encore ResNet50, arrivent en général très bien à identifier les caractéristiques principales d'une image telles que ses contours, angles, variations de couleur, etc.

En partant de ce principe (vu en cours) et des conseil de notre assistant Hector, et sachant que l'identification précise de 3 types d'animaux/images seulement nous intéresse vraiment, à savoir *crapouille*, triton, ou autre, les résultats obtenus avec VGG-16 de base n'étaient pas assez précis. Nous arrivions de temps en temps à identifier des tritons, mais jamais de grenouilles ou de crapauds. En général nous obtenions des résultats aléatoires comme des balais à la place des tritons. Nous avons donc supprimé la dernière couche, et l'avons remplacée par notre propre couche entraînée avec les images (un peu plus de 2'600) que nous avons dû relabelliser nous-mêmes avec *Newt*, *Frog (crapaud ou grenouille)* ou *None*, soit 3 sorties possibles seulement.

Les images que nous avons labellisées sont toutes issue des séquences déjà labellisées que nous avons reçues contenant des batraciens.

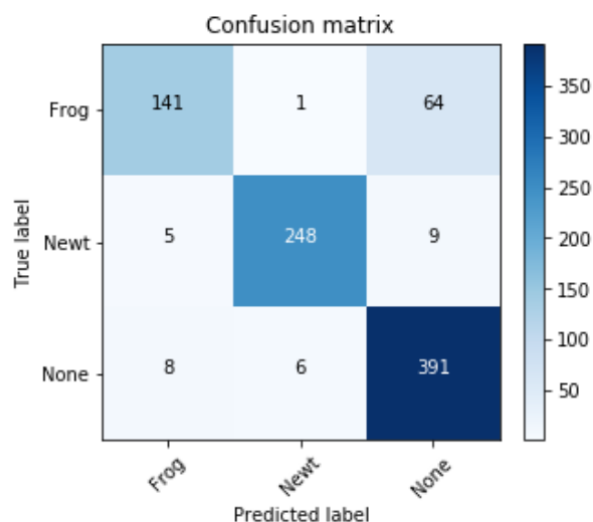
Nous avons aussi pensé à une autre approche, qui aurait été de faire un auto encodeur au lieu de labelliser nous-mêmes les images. Le problème ici, c'est que nous n'avons aucune garantie de résultats ; de plus, instinctivement, nous nous somme plutôt dit que les caractéristiques qui ressortiraient bien seraient principalement la présence ou non de la planche ou de feuilles, et non les animaux. En fait labelliser 2'600 images n'a pas été si long (un après-midi en regardant une série en même temps), vu que les batraciens, si présents, apparaissaient en grand sur nos images.

Pour résumer en deux mots nous avons repris VGG-16 et procédé à du fine-tuning sur la dernière couche du réseau afin de reconnaître uniquement les trois classes qui nous intéressent, et avons donné en entrée à VGG des images ressorties du preprocessing décrit ci-dessus. Mais comme celles-ci sont en noir et blanc, les trois couleurs passées au réseau étaient en réalité trois fois les mêmes, à savoir l'image en nuance de gris (voir plus haut la dernière image des 3 éléments isolés).

## 5. Résultats

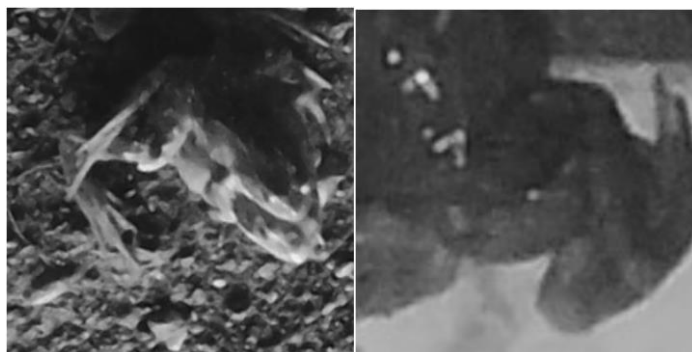
Le réseau a été entraîné avec nos 2600 images labellisées, réparties en 66 % pour le train-set et 33 % pour le test set.

Le plus parlant pour montrer nos résultats est de regarder la matrice de confusion ci-dessous :



Matrice de confusion

Nous obtenons ainsi un taux de près de 90 % d'images correctement labellisées, le plus fort taux d'erreur se rapportant à des grenouilles étiquetées comme étant sans batraciens.



Ci-dessus, deux images de *crapouilles* labellisées comme étant "rien" : pour la première, on peut se dire que c'est probablement dû à l'absence de contraste sur certaines parties du couple de *crapouilles*. Ou dans le cas de la deuxième photo, au fait que l'animal n'est pas complet sur la photo.



## 6. Analyse et conclusions

Nous sommes relativement satisfaits des résultats pour un problème aussi complexe, avec aussi peu de contrastes, différences de couleurs, etc. 89 % de réponses correctes semble être un bon résultat. De plus la grande majorité des erreurs sont des prédictions de "rien" au lieu d'un batracien, et encore une fois vu l'objectif final du projet qui est de compter les animaux, il est très probable que ce même animal sera reconnu avec succès dans une autre image de la même séquence .

Nous avons aussi eu des idées pour continuer et encore améliorer le projet, adopter l'une ou l'autre des solutions suivantes ou une combinaison d'entre elles.

1. Au lieu de passer 3 couches RGB de la même image, passer une fois l'image de base, une fois la différence, et une fois le contour, à tester avec des ordres différents.
2. Faire du fine tuning non pas sur la dernière couche de VGG mais sur les 3 dernières.
3. Faire du fine tuning sur les premières couches de VGG vu qu'on n'a pas une image RGB mais 3 images en nuance de gris.

D'autres idées consistent encore à modifier le processus de prétraitement, mais ceci impliquerait de nouveau de labelliser des images à la main. On pourrait par exemple retirer le background de l'image hors du contour, amplifier les images aux emplacements des différences pour ajouter du contraste (une sorte de superposition de l'image et de la différence).

Un autre problème qui surviendra par la suite est "le comptage" d'animaux, en effet chaque animal va apparaître plusieurs fois dans la même séquence. L'idée que nous avons eue pour cette étape est, pour chaque séquence ou séquences proches (à tester), de superposer toutes les différences ; de cette manière on devrait pouvoir voire apparaître un tracé par animal.

Une autre difficulté pour laquelle nous n'avons pas de solution, ni d'idée comment résoudre, c'est les images contenant plusieurs animaux, en particulier les couples de *crapouilles* l'une sur l'autre, ce qui arrive très souvent. On pourrait peut-être essayer d'entraîner le réseau avec une classe supplémentaire qui représenterait les couples.

Enfin, dernière question actuellement sans réponse : comment différencier un crapaud d'une grenouille ? Pour l'instant, seuls les spécialistes en sont capables. Et pour entraîner notre réseau dans ce sens, il faudrait qu'un au moins de ces spécialistes labellise de nombreuses images.



---

**Addendum – 19 juin 2018**

Nous avons l'intention de faire tourner une dernière fois notre code pour corriger les derniers détails, obtenir un rendu propre et récupérer quelques copies d'écran, mais il y a malheureusement eu des modifications effectuées sur le serveur. Et ces changements font que notre code ne tourne plus dans son état actuel. Nous préférons bien évidemment ne pas 'réparer' le serveur, pour ne pas risquer de rendre inutilisable le travail d'un de nos camarades de classe. Car effectivement, après vérification, nous avons constaté que tous les groupes ont travaillé dans le même environnement virtuel python/anaconda.