Custom Dataset Creation

1 - Collection of Images

We need images of :

- Cats
- Dogs
- Monkeys

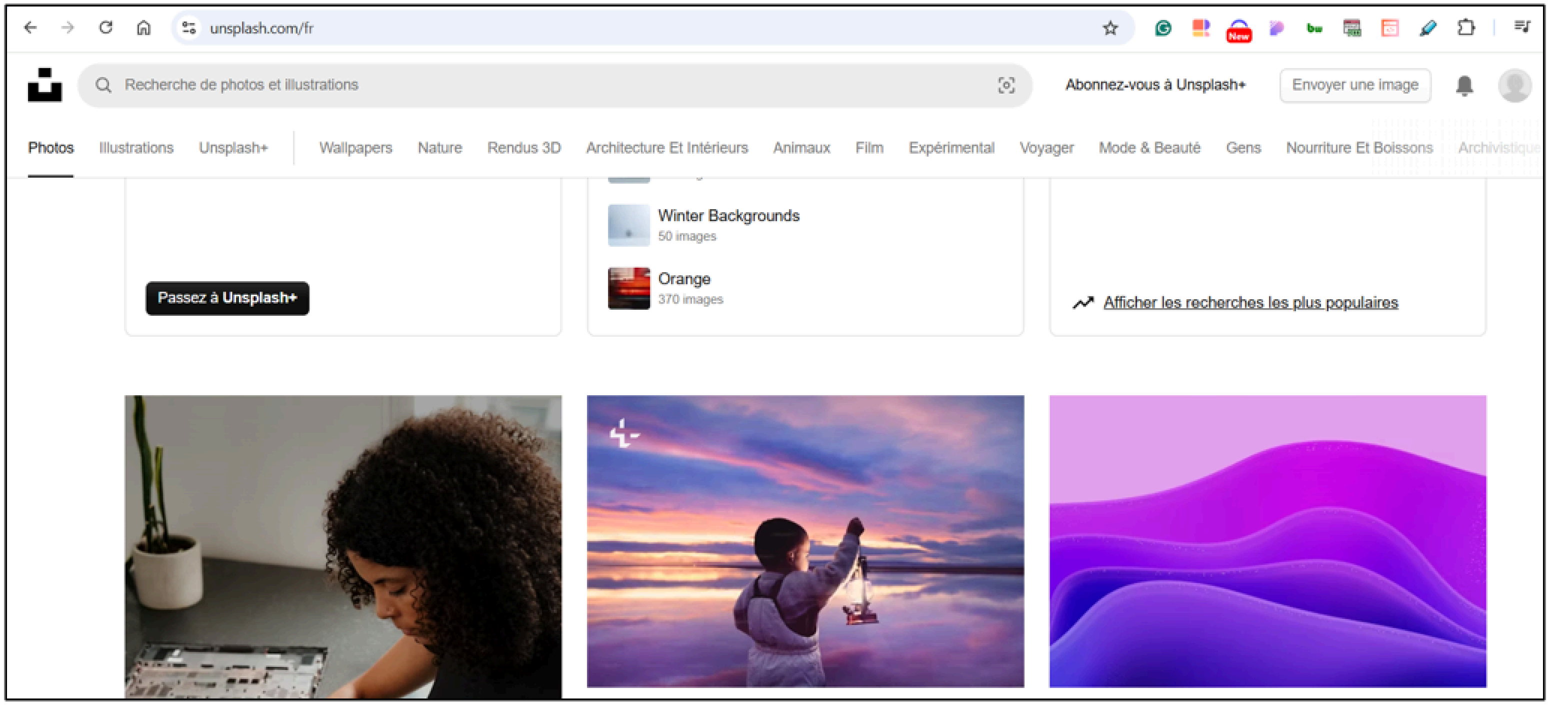
Automating image collection can be achieved through various methods. In this case - we will use **Image APIs** specifically :

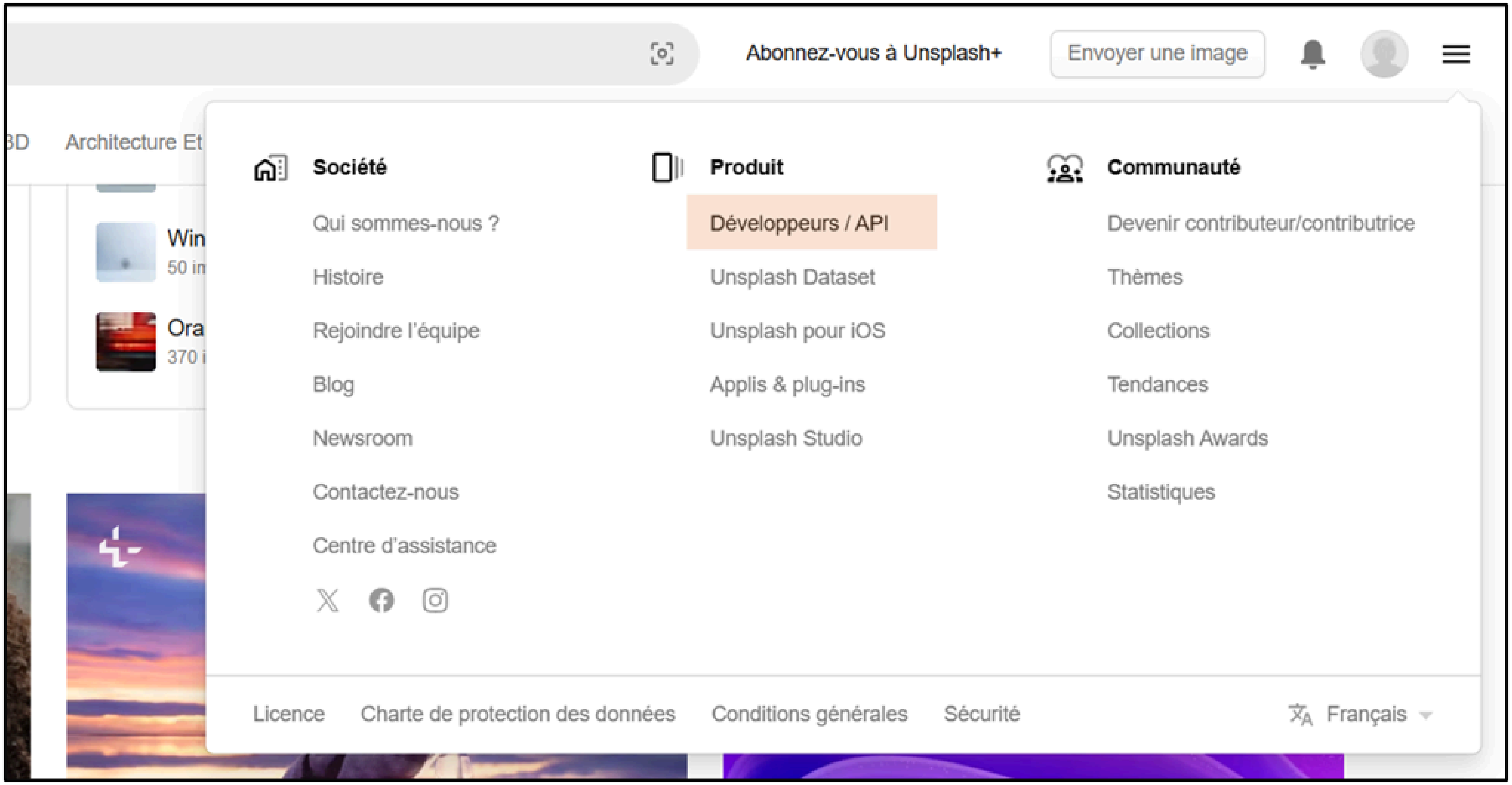
- **Unsplash API**
- **Pexels API**

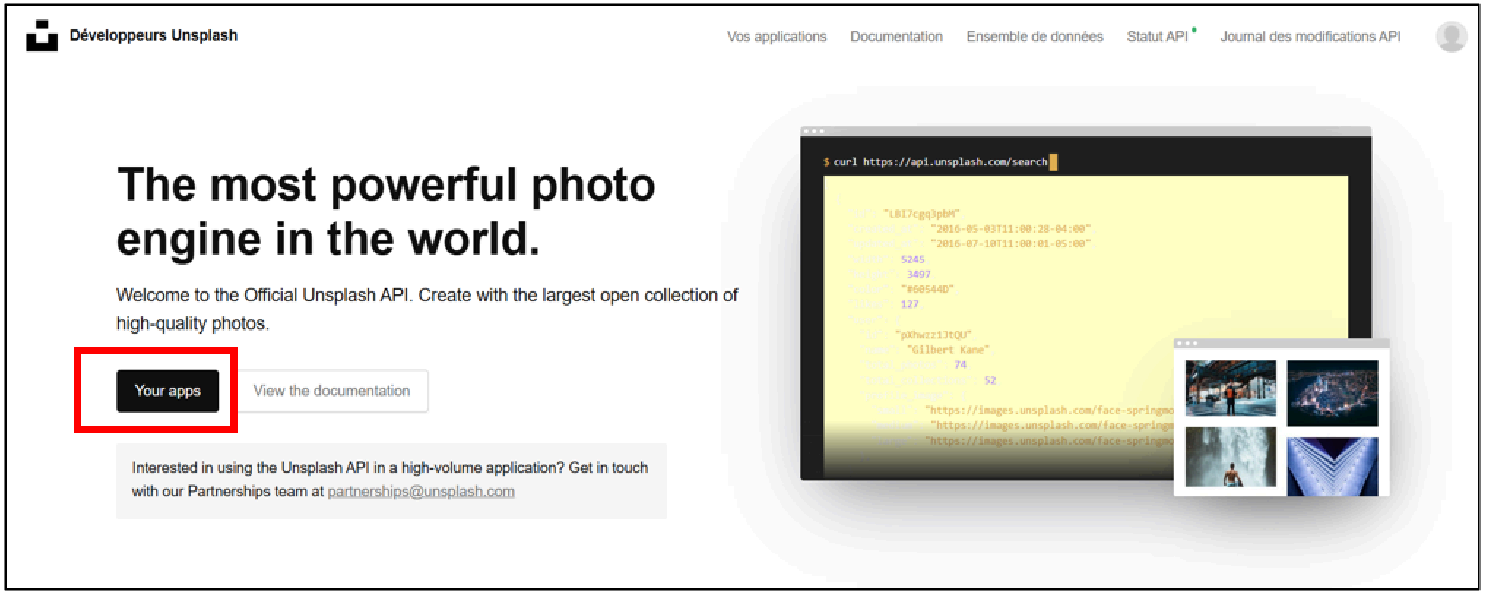
1 - Generate APIs Keys

1 - Create an Account on Unsplash or Pixels

2 - Get the Keys


unsplash.com/fr
Recherche de photos et illustrations
Abonnez-vous à Unsplash+
Envoyer une image
Photos
Illustrations
Unsplash+
Wallpapers
Nature
Rendus 3D
Architecture Et Intérieurs
Animaux
Film
Expérimental
Voyager
Mode & Beauté
Gens
Nourriture Et Boissons
Archivistique
Winter Backgrounds
50 images
Orange
370 images
Passez à Unsplash+
Afficher les recherches les plus populaires


Abonnez-vous à Unsplash+
Envoyer une image
3D
Architecture Et
Win
50 im
Ora
370 i
Société
Qui sommes-nous ?
Histoire
Rejoindre l'équipe
Blog
Newsroom
Contactez-nous
Centre d'assistance
Produit
Développeurs / API
Unsplash Dataset
Unsplash pour iOS
Applis & plug-ins
Unsplash Studio
Communauté
Devenir contributeur/contributrice
Thèmes
Collections
Tendances
Unsplash Awards
Statistiques
Licence
Charte de protection des données
Conditions générales
Sécurité
Français

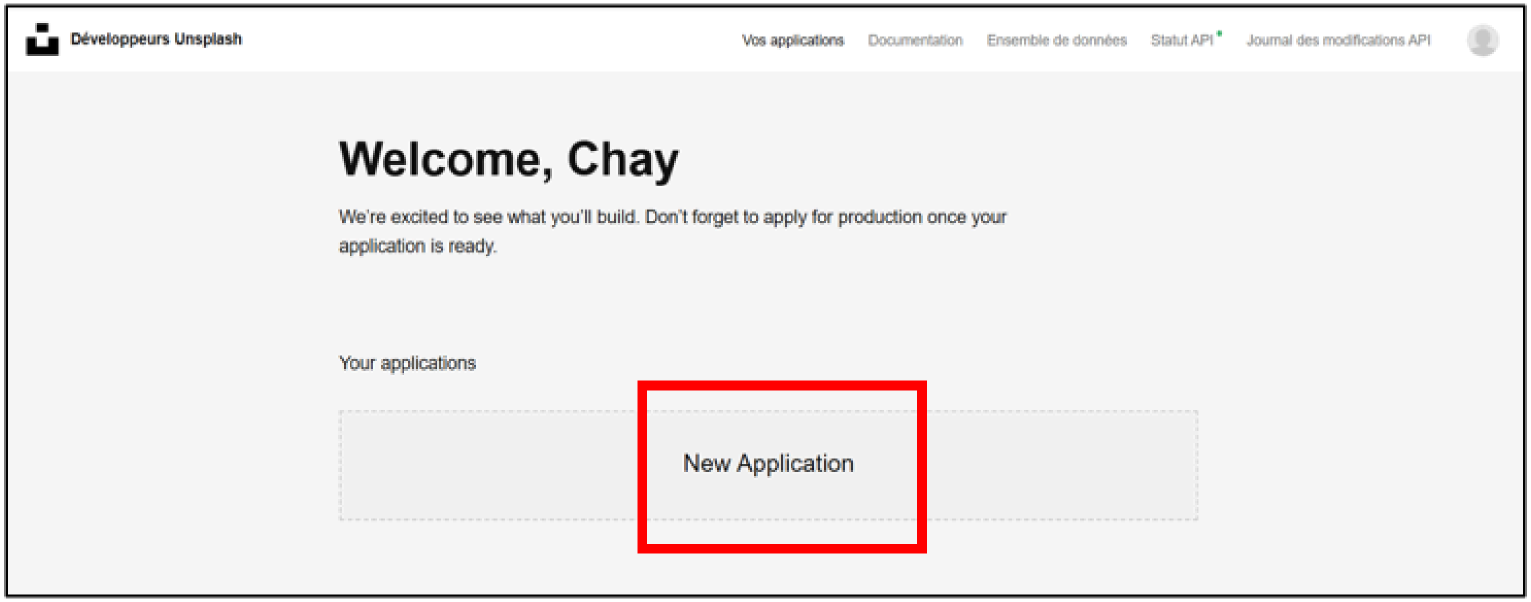


The most powerful photo engine in the world.

Welcome to the Official Unsplash API. Create with the largest open collection of high-quality photos.

Interested in using the Unsplash API in a high-volume application? Get in touch with our Partnerships team at partnerships@unsplash.com

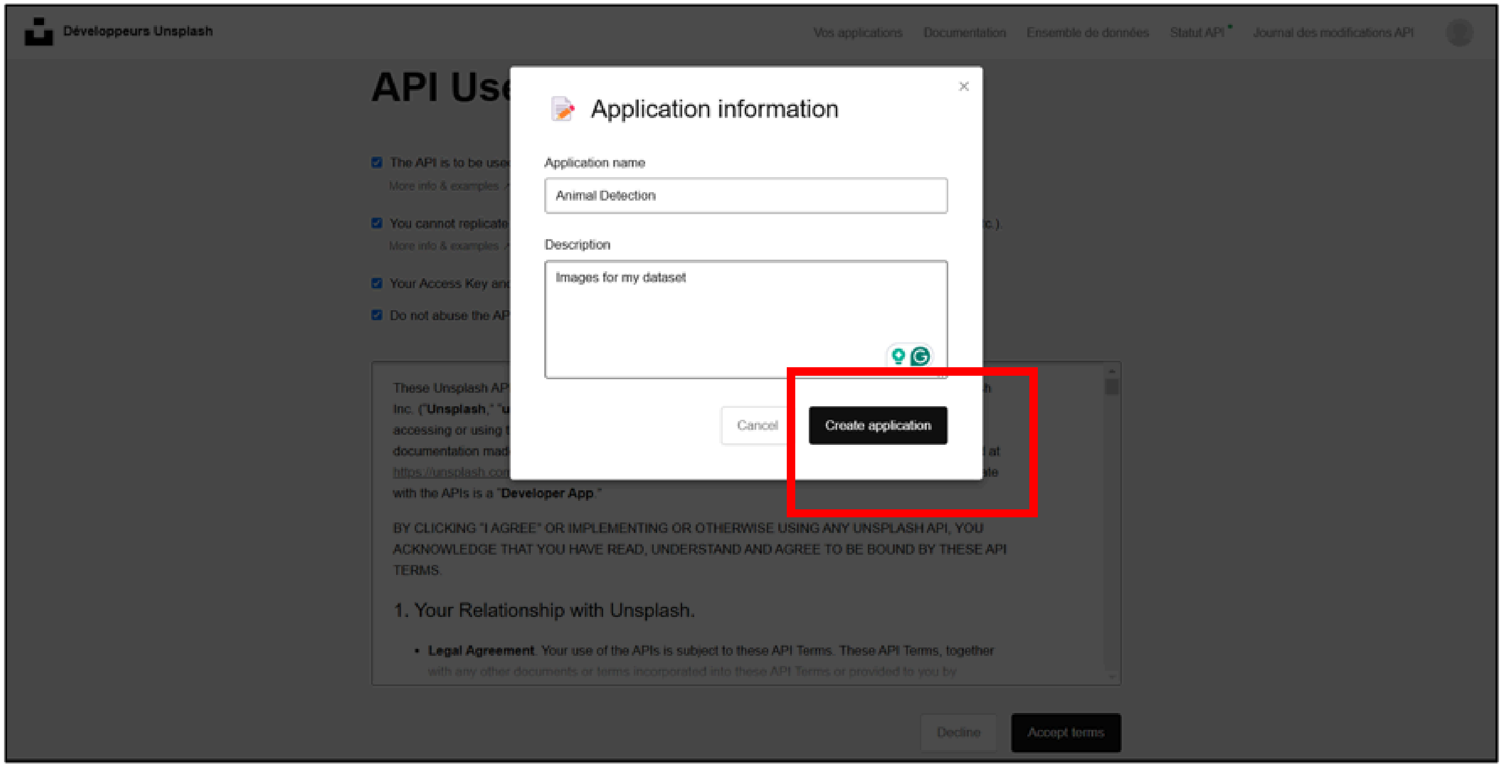
```
$ curl https://api.unsplash.com/search
[
  {
    "id": "LBI7cgq3pbM",
    "created_at": "2016-05-03T11:00:28-04:00",
    "updated_at": "2016-07-10T11:00:01-05:00",
    "width": 5245,
    "height": 3497,
    "color": "#60544D",
    "likes": 127,
    "user": {
      "id": "pXhwzz1JtQU",
      "name": "Gilbert Kane",
      "total_photos": 74,
      "total_collections": 52,
      "profile_image": {
        "small": "https://images.unsplash.com/face-springmo
        "medium": "https://images.unsplash.com/face-springm
        "large": "https://images.unsplash.com/face-springmo
      },
```



Welcome, Chay

We're excited to see what you'll build. Don't forget to apply for production once your application is ready.

Your applications



API Use

☑ The API is to be use

More info & examples

☑ You cannot replicate

More info & examples

☑ Your Access Key an

☑ Do not abuse the AP

Application information

Application name

Animal Detection

Description

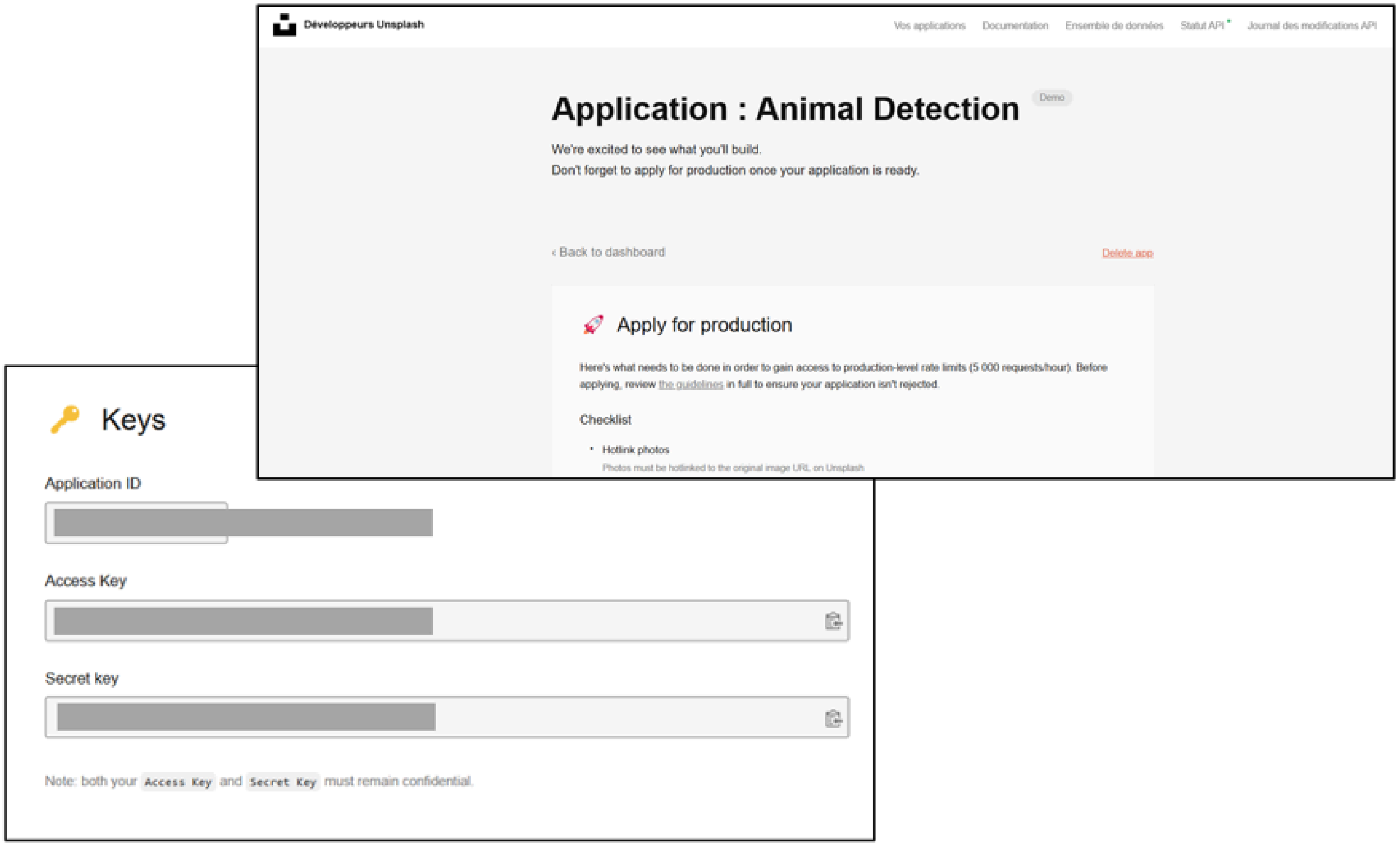
Images for my dataset

These Unsplash AP
Inc. ("**Unsplash**," "u
accessing or using t
documentation mad
https://unsplash.com
with the APIs is a "**Developer App**."

BY CLICKING "I AGREE" OR IMPLEMENTING OR OTHERWISE USING ANY UNSPLASH API, YOU ACKNOWLEDGE THAT YOU HAVE READ, UNDERSTAND AND AGREE TO BE BOUND BY THESE API TERMS.

1. Your Relationship with Unsplash.

- **Legal Agreement**. Your use of the APIs is subject to these API Terms. These API Terms, together with any other documents or terms incorporated into these API Terms or provided to you by



Application : Animal Detection

We're excited to see what you'll build.
Don't forget to apply for production once your application is ready.

Apply for production

Here's what needs to be done in order to gain access to production-level rate limits (5 000 requests/hour). Before applying, review the guidelines in full to ensure your application isn't rejected.

Checklist

- Hotlink photos
  Photos must be hotlinked to the original image URL on Unsplash

Keys

Application ID

[redacted]

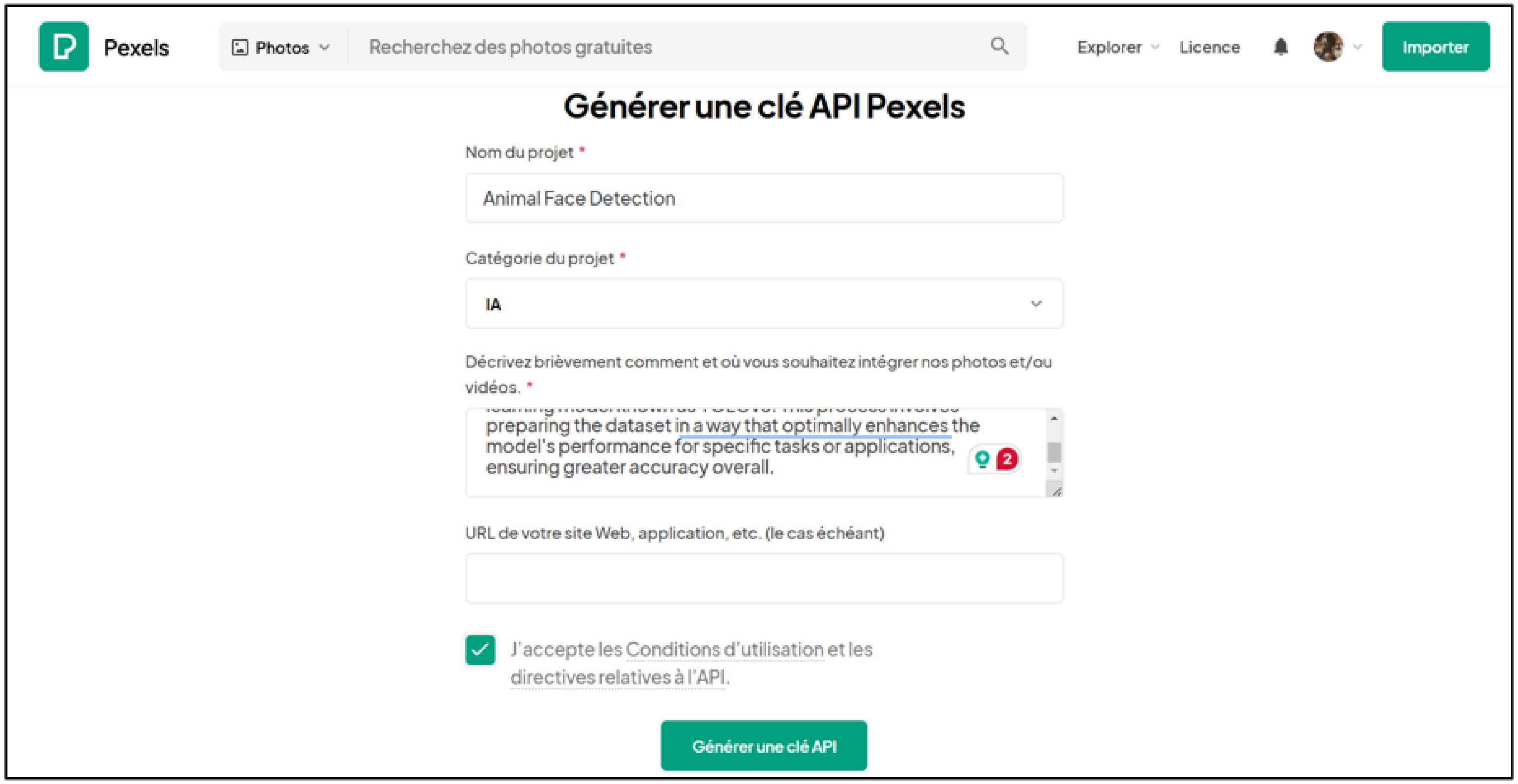
Access Key

[redacted]

Secret key

[redacted]

Note: both your `Access Key` and `Secret Key` must remain confidential.



Générer une clé API Pexels

Nom du projet *

Animal Face Detection

Catégorie du projet *

IA

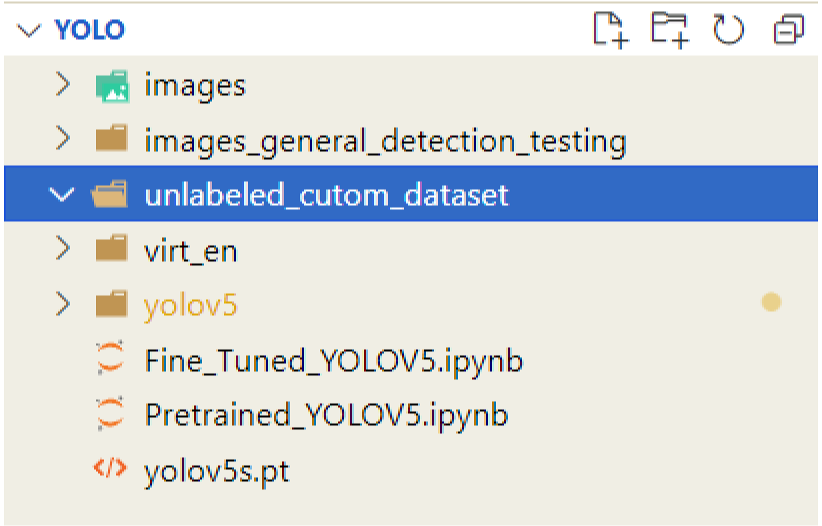
Décrivez brièvement comment et où vous souhaitez intégrer nos photos et/ou vidéos. *

preparing the dataset in a way that optimally enhances the model's performance for specific tasks or applications, ensuring greater accuracy overall.

URL de votre site Web, application, etc. (le cas échéant)

☑ J'accepte les Conditions d'utilisation et les directives relatives à l'API.

2 - Create the folder - unlabeled_custom_dataset - fro storing the fetched Images



YOLO

- images
- images_general_detection_testing
- unlabeled_cutom_dataset
- virt_en
- yolov5
- Fine_Tuned_YOLOV5.ipynb
- Pretrained_YOLOV5.ipynb
- yolov5s.pt

3 - Fetch and Save Images from Unsplash and Pixels for Cats-Dogs-Monkeys

In [2]:

```python
import requests # type: ignore
import os
```

In [3]:

```python
# Unsplash Access Key
access_key_unsplash = '5M9gvUyY9FNunePtJ_9KSn64FY8hpZUv-wlgtoVjKGo'
access_key_pixels = 'gNl8yOkJ3sbKnEsI0PseobgLIrNCG4vC7yj0QXRG1Qh98Vq7xErg0jCU'
```

Unsplash

In [3]:

```python
downloaded_urls = set()  # Set to store downloaded image URLs
```

In [10]:

```python
# Function to fetch and save images from Unsplash
def fetch_unsplash_images(query, num_images, access_key, folder='unlabeled_custom_dataset'):
    images_fetched = 0

    # Unsplash allows up to 30 images per request
    batch_size = 30

    # Ensure the folder exists
    os.makedirs(folder, exist_ok=True)


    while images_fetched < num_images:
        # Calculate remaining images to download in the current batch
        current_batch = min(batch_size, num_images - images_fetched)

        # API request to fetch random images
        url = f"https://api.unsplash.com/photos/random?query={query}&count={current_batch}&client_id={access_key}"
        response = requests.get(url)

        if response.status_code == 200:
            images = response.json()
            for i, image in enumerate(images):
                img_url = image['urls']['full']  # Full-resolution image URL

                # Skip if the image URL is already downloaded
                if img_url in downloaded_urls:
                    continue

                # Mark this image URL as downloaded
                downloaded_urls.add(img_url)

                # Fetch the image data
                img_data = requests.get(img_url).content

                # Save the image to the specified folder
                img_name = f"dog_{images_fetched + i + 1}.jpg"
                with open(os.path.join(folder, img_name), 'wb') as file:
                    file.write(img_data)

            images_fetched += current_batch
            print(f"Downloaded {images_fetched}/{num_images} images.")
        else:
            print(f"Error: {response.status_code}. Exiting.")
            break
```

In [ ]:

```
# Call the function to fetch images of cats
fetch_unsplash_images('dog', 1100, access_key_unsplash)
```

Pixels

In [5]:

```python
def download_images_from_pexels(api_key, query, num_images, folder_path='unlabeled_custom_dataset'):
    # Pexels API URL
    API_URL = 'https://api.pexels.com/v1/search'
    PER_PAGE = 80  # Maximum allowed value for 'per_page' by the Pexels API

    # Ensure the folder exists
    if not os.path.exists(folder_path):
        os.makedirs(folder_path)

    # API request headers
    headers = {
        "Authorization": api_key
    }

    total_downloaded = 0
    page = 1

    while total_downloaded < num_images:
        remaining_images = num_images - total_downloaded
        per_page = min(PER_PAGE, remaining_images)

        # Fetch images from Pexels
        response = requests.get(API_URL, headers=headers, params={
            "query": query,
            "per_page": per_page,
            "page": page
        })

        if response.status_code == 200:
            data = response.json()
            photos = data.get("photos", [])

            if not photos:  # Break if no more photos are available
                print("No more images available.")
                break

            for photo in photos:
                total_downloaded += 1
                image_url = photo["src"]["original"]
                image_response = requests.get(image_url)
                if image_response.status_code == 200:
                    image_path = os.path.join(folder_path, f"{query}{total_downloaded}.jpg")
                    with open(image_path, "wb") as file:
                        file.write(image_response.content)
                    print(f"Downloaded: {image_path}")
                else:
                    print(f"Failed to download image {total_downloaded}")

                if total_downloaded >= num_images:
                    break

            page += 1
        else:
            print(f"Failed to fetch data: {response.status_code} - {response.text}")
            break
```

In [ ]:

```
# Call the function to fetch images of cats
download_images_from_pexels
(access_key_pixels, 'cat', 550)
```

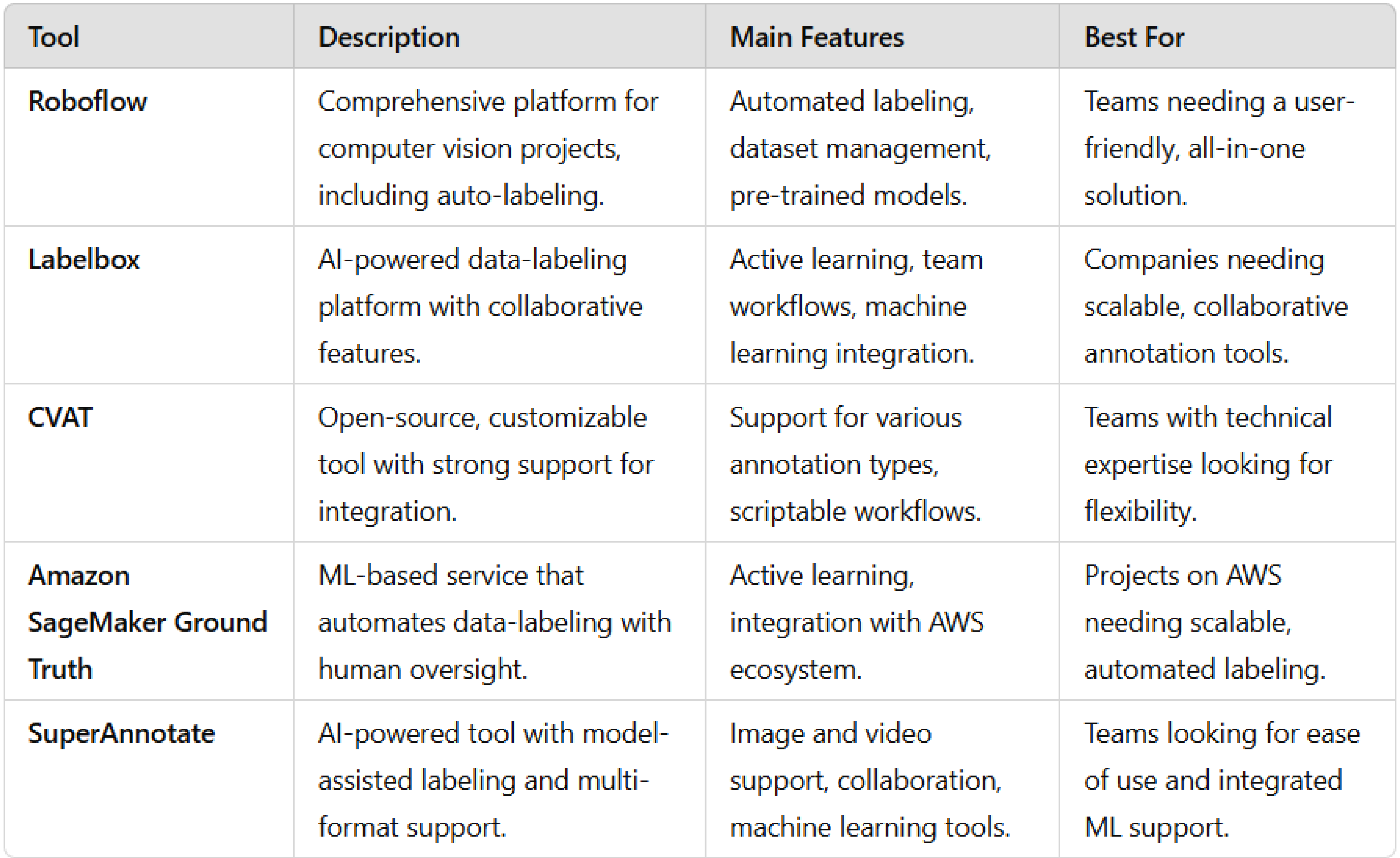
3 - Cleaning of Images

3 - Annotation of Images

1 - Annotating a Subset of unlabeled_custom_dataset

We will select **100** images from each **class** in our unlabeled_custom_dataset for annotation.

Some of the popular Annotation Tools are :

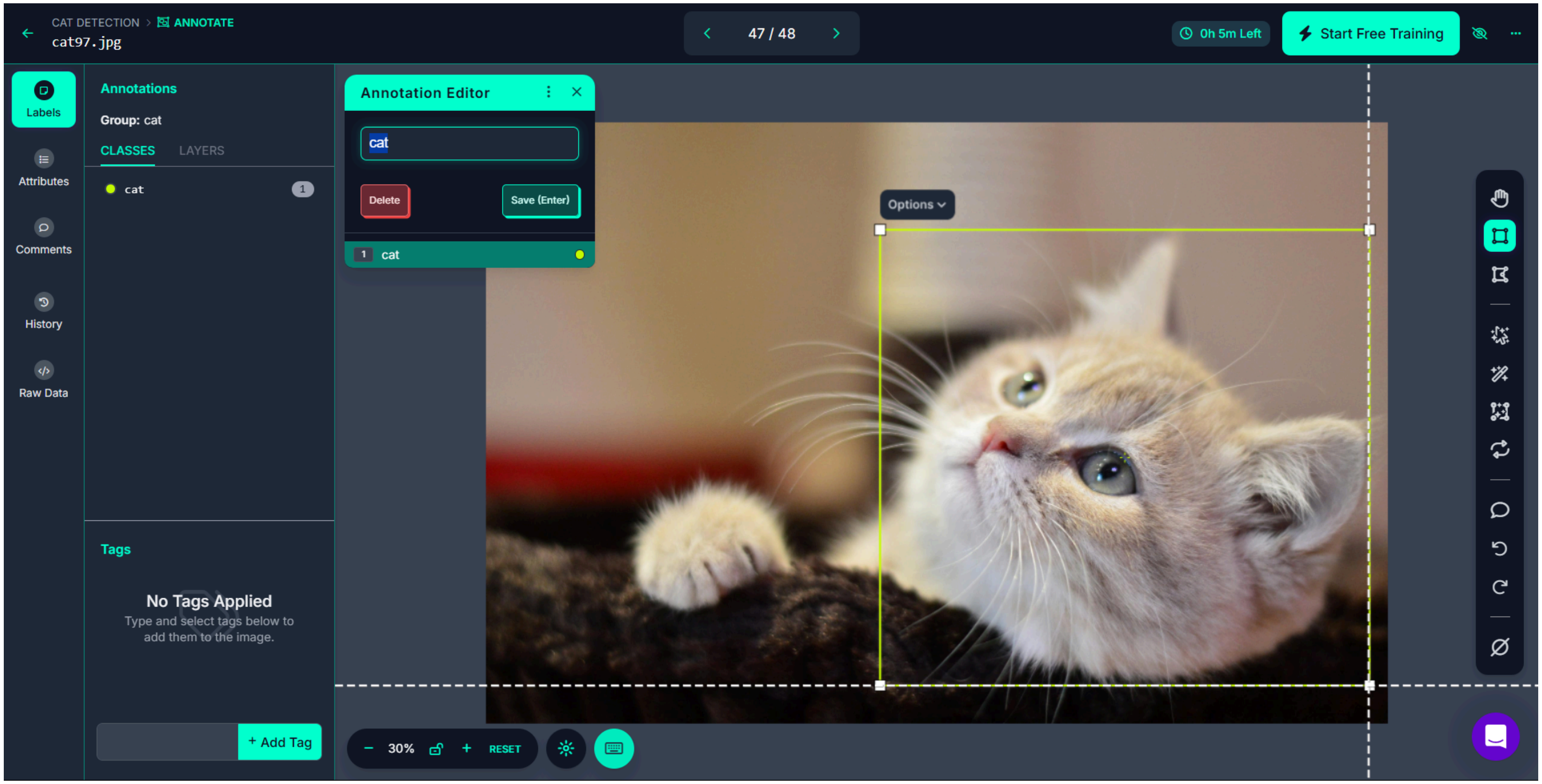


Tool	Description	Main Features	Best For
Roboflow	Comprehensive platform for computer vision projects, including auto-labeling.	Automated labeling, dataset management, pre-trained models.	Teams needing a user-friendly, all-in-one solution.
Labelbox	AI-powered data-labeling platform with collaborative features.	Active learning, team workflows, machine learning integration.	Companies needing scalable, collaborative annotation tools.
CVAT	Open-source, customizable tool with strong support for integration.	Support for various annotation types, scriptable workflows.	Teams with technical expertise looking for flexibility.
Amazon SageMaker Ground Truth	ML-based service that automates data-labeling with human oversight.	Active learning, integration with AWS ecosystem.	Projects on AWS needing scalable, automated labeling.
SuperAnnotate	AI-powered tool with model-assisted labeling and multi-format support.	Image and video support, collaboration, machine learning tools.	Teams looking for ease of use and integrated ML support.

Roboflow

Roboflow is a popular **platform** designed to simplify and streamline the process of **building and deploying computer vision models**.
It offers a suite of **tools** that cater to the needs of data scientists - machine learning engineers and developers working on **computer vision projects**.

roboflow



No Tags Applied
Type and select tags below to add them to the image.

4 - Data Augmentation

5 - Preprocess of the Dataset

1 - Resizing

2 - Normalisation

In general - letting YOLOv5 handle both resizing and normalization during the training process is the recommended approach.