Welcome to AniML.

Computer Vision made easy for automated image processing.

AniMLs is an easy tool for the rapid creation of computer vision systems for analyzing and filtering a large set of raw images down to images of interest

AniML - Home Page

1. Clean up your dataset

   Take control of a chaotic dataset by uploading it to the platform - AniML automatically sorts and finds images with objects of interest, which you will later label and use to train a supervised ML model.

2. Train your own AI vision system

   Using the filtered images from step 1, add labels detailing the class and location of each object you're trying to detect. Using the labeled dataset, train your own custom ML model that will automatically detect objects of interest for you.

3. Use your vision system to find objects and generate insights

   With your new model, automate away all your previously manual steps! Upload new images and AniML will automatically find images with objects of interest AND generate a comprehensive dashboards of metrics detailing what the model found.

Creators: Ivan Wong, Lana Elauria, Lucas Harvey-Schroyer, Whit Blodgett

UC Berkeley, School of Information 2022 - Masters of Information and Data Science

Automatic Image Filtering

No more combing through thousands of false triggers! Add all your images below and we will filter out the images that contain objects of interest from those that are empty or don't contain any objects of interest (e.g. a car).

AniML - Model 1. Identify and Filter Empty from Animal Images

Upload your file with all the images you've captured and AniML will find filter down to the images of interest i.e. those with objects of interest that you normally manually search for. Once added, AniML runs these image through a standard, pre-trained model that will act as a sort of anomaly detection engine, surfacing up any images that contain an objects it recognizes. Don't worry if the classifications are inaccurate, we'll soon train a hyper custom model uniquely tuned to your case. Next you can click the "Review Filtered Image" tab to view and save the filtered results.

Next you can click the "Review Filtered Image" tab to view and save the filtered results.

Choose Files No file chosen

Upload and Process

Automatic Image Filtering

No more combing through thousands of false triggers! Add all your images below and we will filter out the images that contain objects of interest from those that are empty or don't contain any objects of interest (e.g. a car).

AniML - Model 1. Identify and Filter Empty from Animal Images

Upload your file with all the images you've captured and AniML will find filter down to the images of interest i.e. those with objects of interest that you normally manually search for. Once added, AniML runs these image through a standard, pre-trained model that will act as a sort of anomaly detection engine, surfacing up any images that contain an objects it recognizes. Don't worry if the classifications are inaccurate, we'll soon train a hyper custom model uniquely tuned to your case. Next you can click the "Review Filtered Image" tab to view and save the filtered results.

Next you can click the "Review Filtered Image" tab to view and save the filtered results.

Choose Files 18 files

Upload and Process

[
  {
    "x": "0.76622736",
    "y": "0.7959806",
    "w": "0.4641934",
    "h": "0.33204788",
    "prob": "0.5066631",
    "class": "cat"
  }
]

Click any of the view buttons below to view individual filtering model results here.

Task Id	Status	Initial Model Filtering Prediction	Action
40e7d3b7-6d6f-42ae-985c-b1ae41530026	SUCCESS	View	{"task_id":"40e7d3b7-6d6f-42ae-985c-b1ae41530026","status":"PROCESSING","url_result":"/api/result/40e7d3b7-6d6f-42ae-985c b1ae41530026"}
f972e056-378a-4a3f-ada5-f0c5501261b6	SUCCESS	View	{"task_id":"f972e056-378a-4a3f-ada5-f0c5501261b6","status":"PROCESSING","url_result":"/api/result/f972e056-378a-4a3f-ada5-f0c5501261b6"}
9012a114-34a7-4c19-9b42-aafd63571f2b	SUCCESS	View	{"task_id":"9012a114-34a7-4c19-9b42-aafd63571f2b","status":"PROCESSING","url_result":"/api/result/9012a114-34a7-4c19-9b42 aafd63571f2b"}

Task Id	Status	Initial Model Filtering Prediction	Action
ef75bd05-4d5d-4307-bbe9-677b2d53b0ed	SUCCESS	View	{"task_id":"ef75bd05-4d5d-4307-bbe9-677b2d53b0ed","status":"PROCESSING","url_result":"/api/result/ef75bd05-4d5d-4307-bbe 677b2d53b0ed"}
1b73ce2d-a7d7-402c-8566-41d639848b85	SUCCESS	View	{"task_id":"1b73ce2d-a7d7-402c-8566-41d639848b85","status":"PROCESSING","url_result":"/api/result/1b73ce2d-a7d7-402c-856 41d639848b85"}
79501e58-038c-4276-a9ba-1ddf92539a37	SUCCESS	View	{"task_id":"79501e58-038c-4276-a9ba-1ddf92539a37","status":"PROCESSING","url_result":"/api/result/79501e58-038c-4276-a9ba 1ddf92539a37"}
f630e866-d004-4a4b-8a5b-65ede79ff1db	SUCCESS	View	{"task_id":"f630e866-d004-4a4b-8a5b-65ede79ff1db","status":"PROCESSING","url_result":"/api/result/f630e866-d004-4a4b-8a5b 65ede79ff1db"}
c885370a-33ac-4a7d-9058-a2c240d423a3	SUCCESS	View	{"task_id":"c885370a-33ac-4a7d-9058-a2c240d423a3","status":"PROCESSING","url_result":"/api/result/c885370a-33ac-4a7d-9058 a2c240d423a3"}
6e2eb257-24ec-4efd-8f1e-7e607905e087	SUCCESS	View	{"task_id":"6e2eb257-24ec-4efd-8f1e-7e607905e087","status":"PROCESSING","url_result":"/api/result/6e2eb257-24ec-4efd-8f1e 7e607905e087"}
833fb268-6ea3-444e-a10e-6ed9a088322c	SUCCESS	View	{"task_id":"833fb268-6ea3-444e-a10e-6ed9a088322c","status":"PROCESSING","url_result":"/api/result/833fb268-6ea3-444e-a10e 6ed9a088322c"}
8bfee8f8-ccf0-4715-82ba-26fdb3b67082	SUCCESS	View	{"task_id":"8bfee8f8-ccf0-4715-82ba-26fdb3b67082","status":"PROCESSING","url_result":"/api/result/8bfee8f8-ccf0-4715-82ba-26fdb3b67082"}
f9374b4d-a1d0-43ac-9038-4c6cedb91042	SUCCESS	View	{"task_id":"f9374b4d-a1d0-43ac-9038-4c6cedb91042","status":"PROCESSING","url_result":"/api/result/f9374b4d-a1d0-43ac-9038 4c6cedb91042"}
3cb48848-fae0-43b2-88f0-531a77bb7c54	SUCCESS	View	{"task_id":"3cb48848-fae0-43b2-88f0-531a77bb7c54","status":"PROCESSING","url_result":"/api/result/3cb48848-fae0-43b2-88f0 531a77bb7c54"}
8bed7f73-043b-4d42-a2de-7c808f436cd3	SUCCESS	View	{"task_id":"8bed7f73-043b-4d42-a2de-7c808f436cd3","status":"PROCESSING","url_result":"/api/result/8bed7f73-043b-4d42-a2de 7c808f436cd3"}
d76b4b7c-ffaf-4994-856f-2dff165d4028	SUCCESS	View	{"task_id":"d76b4b7c-ffaf-4994-856f-2dff165d4028","status":"PROCESSING","url_result":"/api/result/d76b4b7c-ffaf-4994-856f-2dff165d4028"}
a92b8bbb-ab97-4170-aee3-b4e04fb01185	SUCCESS	View	{"task_id":"a92b8bbb-ab97-4170-aee3-b4e04fb01185","status":"PROCESSING","url_result":"/api/result/a92b8bbb-ab97-4170-aee b4e04fb01185"}
febcaa7f-e3a9-4e9b-a05f-3b2d89c74013	SUCCESS	View	{"task_id":"febcaa7f-e3a9-4e9b-a05f-3b2d89c74013","status":"PROCESSING","url_result":"/api/result/febcaa7f-e3a9-4e9b-a05f-3b2d89c74013"}
fcbcd877-2b96-4ff2-b2ac-74300593b20f	SUCCESS	View	{"task_id":"fcbcd877-2b96-4ff2-b2ac-74300593b20f","status":"PROCESSING","url_result":"/api/result/fcbcd877-2b96-4ff2-b2ac-74300593b20f"}

Review Image Classifications

Here we can see the results from our filtering model's prediction of which images are false triggers. Adjust the sensitivity and randomly view a sample of the images.

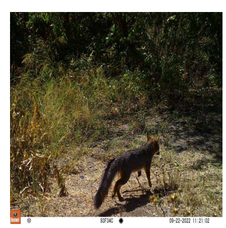
AniML - Filtering Empty Images

Image Classification Results

This table shows a random sample of images that the filtering model predicted as containing and not containing animals. To view more images in each class to get a sense of how the filterin model is performing, click the 'randomize images by class' button. You can adjust the sensitivity of how a model classifies an image as containing an animal or not in the next section below. When you are ready, you can save a zip file of your filtered results by clicking the 'Download zip' button at the top of either column.

Randomly Sampled Images by Predicted Class

randomize images by class

Initial Model Predictions	Initial Model Predictions
Low likelihood image contains animal	**High likelihood image contains animal**
download zip of prediction of no animal images	download zip of prediction of animal images
 09240173_JPG.rf.23abc6723cadc41684176f238fafccd78.jpg	 09200088_JPG.rf.f087b7a8a23bf745b932d1da2e4b2a93.jpg
 09250142_JPG.rf.79abmab7835abalilb3147fafrb1kfaof.jpg	 09110025_JPG.rf.9490847dd182c94e5cedb8fa5cf86500.jpg
 10020175_JPG.rf.78dbaf78afdlkjcacd04f4789dfcdd34d.jpg	

Adjust Image Classification Results

The model classifies an animal as detected (i.e. "animal_detected") or not detected (i.e. "not_animal_detected") based on the confidence level that the image belongs to that class. For "animal_detected", a higher "confidence_score" in the table means the model is more confident that it contains an animal, while for "not_animal_detected", a lower "confidence_score" means that the model is less confident that an animal isn't in the image.

You may adjust the confidence thresholds for each class and see in the image results above how different confidence level thresholds affect the sensitivity of the model predictions above. You may save model results iteratively for different confidence scenarios, such as only saving results where the model has a high confidence that an animal isn't present (i.e. a high confidence cutoff for "not_animal_detected"), a low confidence that an animal is present (i.e. a low confidence cutoff for "animal_detected"), a combination of the two levers, or some other scenario.

If you wish to view the inital model predictions, you can reset the table below.

Reclassify with new confidence level

Keep class if confidence score greater than:

Select Class (i.e. 'animal_detected' or 'not_animal_detected') and New Confidence Cutoff Level:

select from dropdown

e.g. 0.50

Submit

Confidence Score Cutoffs

animal_detected	not_animal_detected
default	default

Reset classification

Reset the 'new' classifications to the default model outputs

Reset classifications

Initial Model Results

	image_ids	confidence_scores	predicted_classes_original
0	09220174_JPG.rf.d67daflj41bncei1814cdlwofuda023147cfw	0.78120	not_animal_detected
1	10020175_JPG.rf.78dbaf78afdlkjcacd04f4789dfcdd34d	0.69390	not_animal_detected
2	10020177_JPG.rf.89dfd98fdfef898421cd3983cfe238adf	0.88130	not_animal_detected
3	09080004_JPG.rf.867d658be6573c5b18f3a26467744e7b	0.72353	not_animal_detected
4	09080005_JPG.rf.adcfe72715aaac8a3bfafb6dd0cb4d3c	0.63789	not_animal_detected
5	09220107_JPG.rf.1177492fb393be327df414b8180642db	0.84609	animal_detected
6	09220112_JPG.rf.82197ddbc728a10e41d72941f8af6116	0.71150	animal_detected
7	09180073_JPG.rf.7448f1e657b45ef54b66b72703577fa4	0.61910	animal_detected
8	09220113_JPG.rf.67904d0764563fd6d921e907e034ab23	0.53751	animal_detected
9	09220114_JPG.rf.d3671bfbdda727793746fdfbd12b7371	0.37580	animal_detected
10	09200088_JPG.rf.f087b7a8a23bf745b932d1da2e4b2a93	0.62989	animal_detected
11	09260140_JPG.rf.92ae3de09b0a1e9f9353d7b07c814641	0.92311	animal_detected
12	09260141_JPG.rf.a95ea71af019915331d8918030a7ec9a	0.89160	animal_detected
13	09230121_JPG.rf.be314214a792202918e4028ec5ec71e7	0.44693	animal_detected
14	09180079_JPG.rf.b655f8164e6875d1c2479aa844f757e8	0.70681	animal_detected
15	09090019_JPG.rf.b8def9071a90f5525f050cb0e0d892f9	0.87569	animal_detected
16	09090022_JPG.rf.ac8543bc7813616d7b3598404f6e3470	0.20554	not_animal_detected

Review Image Classifications

Here we can see the results from our filtering model's prediction of which images are false triggers. Adjust the sensitivity and randomly view a sample of the images.

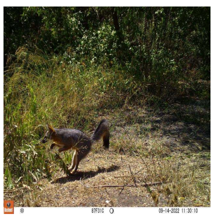
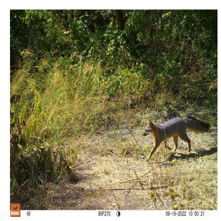
AniML - Filtering Empty Images

Image Classification Results

This table shows a random sample of images that the filtering model predicted as containing and not containing animals. To view more images in each class to get a sense of how the filterin model is performing, click the 'randomize images by class' button. You can adjust the sensitivity of how a model classifies an image as containing an animal or not in the next section below. When you are ready, you can save a zip file of your filtered results by clicking the 'Download zip' button at the top of either column.

Randomly Sampled Images by Predicted Class

randomize images by class

Adjusted Model Predictions	Adjusted Model Predictions
Low likelihood image contains animal	**High likelihood image contains animal**
download zip of prediction of no animal images	download zip of prediction of animal images
 09140037_JPG.rf.b826ac1744e01dd10edf59d212e143b1.jpg	 09090019_JPG.rf.b8def9071a90f5525f050cb0e0d892f9.jpg
 10020177_JPG.rf.89dfd98fdfef898421cd3983cfe238adf.jpg	 09160055_JPG.rf.07b029d123229fd2e960b688ca24eaac.jpg
	

Adjust Image Classification Results

The model classifies an animal as detected (i.e. "animal_detected") or not detected (i.e. "not_animal_detected") based on the confidence level that the image belongs to that class. For "animal_detected", a higher "confidence_score" in the table means the model is more confident that it contains an animal, while for "not_animal_detected", a lower "confidence_score" means that the model is less confident that an animal isn't in the image.

You may adjust the confidence thresholds for each class and see in the image results above how different confidence level thresholds affect the sensitivity of the model predictions above. You may save model results iteratively for different confidence scenarios, such as only saving results where the model has a high confidence that an animal isn't present (i.e. a high confidence cutoff for "not_animal_detected"), a low confidence that an animal is present (i.e. a low confidence cutoff for "animal_detected"), a combination of the two levers, or some other scenario.

If you wish to view the inital model predictions, you can reset the table below.

Reclassify with new confidence level

Keep class if confidence score greater than:

Select Class (i.e. 'animal_detected' or 'not_animal_detected') and New Confidence Cutoff Level:

select from dropdown

e.g. 0.50

Submit

Confidence Score Cutoffs

animal_detected	not_animal_detected
0.800000	default

Reset classification

Reset the 'new' classifications to the default model outputs

Reset classifications

Adjusted Model Results

	image_ids	confidence_scores	predicted_classes_original	predicted_classes_new
0	09220174_JPG.rf.d67daflj41bncei1814cdlwofuda023147cfw	0.78120	not_animal_detected	not_animal_detected
1	10020175_JPG.rf.78dbaf78afdlkjcacd04f4789dfcdd34d	0.69390	not_animal_detected	not_animal_detected
2	10020177_JPG.rf.89dfd98fdfef898421cd3983cfe238adf	0.88130	not_animal_detected	not_animal_detected
3	09080004_JPG.rf.867d658be6573c5b18f3a26467744e7b	0.72353	not_animal_detected	not_animal_detected
4	09080005_JPG.rf.adcfe72715aaac8a3bfafb6dd0cb4d3c	0.63789	not_animal_detected	not_animal_detected
5	09220107_JPG.rf.1177492fb393be327df414b8180642db	0.84609	animal_detected	animal_detected
6	09220112_JPG.rf.82197ddbc728a10e41d72941f8af6116	0.71150	animal_detected	not_animal_detected
7	09180073_JPG.rf.7448f1e657b45ef54b66b72703577fa4	0.61910	animal_detected	not_animal_detected
8	09220113_JPG.rf.67904d0764563fd6d921e907e034ab23	0.53751	animal_detected	not_animal_detected
9	09220114_JPG.rf.d3671bfbdda727793746fdfbd12b7371	0.37580	animal_detected	not_animal_detected
10	09200088_JPG.rf.f087b7a8a23bf745b932d1da2e4b2a93	0.62989	animal_detected	not_animal_detected
11	09260140_JPG.rf.92ae3de09b0a1e9f9353d7b07c814641	0.92311	animal_detected	animal_detected
12	09260141_JPG.rf.a95ea71af019915331d8918030a7ec9a	0.89160	animal_detected	animal_detected
13	09230121_JPG.rf.be314214a792202918e4028ec5ec71e7	0.44693	animal_detected	not_animal_detected
14	09180079_JPG.rf.b655f8164e6875d1c2479aa844f757e8	0.70681	animal_detected	not_animal_detected
15	09090019_JPG.rf.b8def9071a90f5525f050cb0e0d892f9	0.87569	animal_detected	animal_detected
16	09090022_JPG.rf.ac8543bc7813616d7b3598404f6e3470	0.20554	not_animal_detected	not_animal_detected

Label Images

If you'd like to go beyond just filtering your model results, use the resources in this page to label your images for custom model training.

AniML - Labeling

Roboflow Introduction

Next you'll need to label your images of interest with bounding boxes - squares which tell the model what you are looking for. We'll use these images to train a custom model that is great at detecting your objects of interest. Simply use the downloaded dataset from (2.2) with the files that contains your animals, go to:

[Roboflow.com]

Please annotate at least 20 images.Once you've labeled, export the image/label zip file and upload it in step 3.1.

Robowflow Annotate

Create an account and follow our custom walkthrough here:

[Roboflow labeling link]

Example Roboflow Bounding Box Annotation

Drag, Drop & Train!

Drag and drop your Labeled Dataset (i.e. Roboflow zip file) & we'll train a custom model that is tailored to your specific animals and setting! Uploaded labeled images will be used to perform custom YOLOv5 model training.

AniML - Training

Upload the zipped 20 +/- labeled images from roboflow and hit the training button to customize our vision AI model to your specific use case. Depending on the number of images you upload, this model will take between 20 min to 2 hours to train so feel free to check back in a while.

Choose Files No file chosen

Train Custom Model

Drag, Drop & Train!

Drag and drop your Labeled Dataset (i.e. Roboflow zip file) & we'll train a custom model that is tailored to your specific animals and setting! Uploaded labeled images will be used to perform custom YOLOv5 model training.

AniML - Training

Upload the zipped 20 +/- labeled images from roboflow and hit the training button to customize our vision AI model to your specific use case. Depending on the number of images you upload, this model will take between 20 min to 2 hours to train so feel free to check back in a while.

Choose Files AniML-Sample-Dataset-1.zip

Training

Task Id	Status	Action
fc24780a-d174-4a02-a033-e2409f673153	PENDING	

Click the view button once model is finished training to view this customized Yolov5 training metadata.

Automatic Object Detection

Congratulations! You've trained a ML vision system that will greatly reduce the amount of manual work required to find insightful information about objects in your images.

AniML - Model 2. Customized Object Detection and Classification

Object Detection

Time to put your custom ML Vision system to work - add new images with your object of interest here and your model will automatically run inference and find the objects you've trained it to detect. You can review the prediction results by selecting "View" and scrolling to the bottom to see the image, the location and the confidence of the prediction. Check out the predictions - if things look good, proceed to 3.2 (review, analytics, and download) or, if predictions are poor, try labeling more images and re-training your model on step 2.2.

Choose Files No file chosen

Upload Images & Classify Objects

Automatic Object Detection

Congratulations! You've trained a ML vision system that will greatly reduce the amount of manual work required to find insightful information about objects in your images.

AniML - Model 2. Customized Object Detection and Classification

Object Detection

Time to put your custom ML Vision system to work - add new images with your object of interest here and your model will automatically run inference and find the objects you've trained to detect. You can review the prediction results by selecting "View" and scrolling to the bottom to see the image, the location and the confidence of the prediction. Check out the predictions - if things look good, proceed to 3.2 (review, analytics, and download) or, if predictions are poor, try labeling more images and re-training your model on step 2.2.

Choose Files 18 files

Upload Images & Classify Objects

[
  {
    "x": "0.559924",
    "y": "0.66946614",
    "w": "0.31854162",
    "h": "0.27010426",
    "prob": "0.79521537",
    "class": "Fox"
  }
]

Click any of the view buttons below to view individual filtering model results here.

Task Id	Status	Action
aa2104f2-b61c-46f7-aa84-efa82188155c	SUCCESS	View
436b4ca9-06da-4346-8ad6-ce14d34d8d41	SUCCESS	View
e1d1cd63-ef96-45ce-b480-7997dbc839bb	SUCCESS	View
ce2e7887-25b7-43b7-8ccf-285b8b47ca37	SUCCESS	View
3645634c-feae-4eb5-a22d-a13e0b1b0856	SUCCESS	View
854801e9-5fc4-4651-b9dd-db483e5b3d03	SUCCESS	View

Task Id	Status	Action
696c4d0b-f434-4581-a96e-4df55c572f02	SUCCESS	View
f4b92bd8-7b8a-4d8d-abdd-64300ecf8d89	SUCCESS	View
0ec2d58c-a20b-4781-820f-67cc5a37b5d2	SUCCESS	View
79728eed-bd68-4dd7-9f99-b019b59fa1d5	SUCCESS	View
fe3f5916-7bab-49ea-adce-8882383161b1	SUCCESS	View
066c8dfd-132a-438e-90ac-76166b5e59b1	SUCCESS	View
4b9c2523-ebe8-4094-be0c-08883f498b5f	SUCCESS	View
93181981-ca20-4e24-a866-19257a3bf7cb	SUCCESS	View
a5b94899-ab00-442a-a37e-d587ef007d9c	SUCCESS	View
a8892533-cb21-47be-acc6-f8bfb0085100	SUCCESS	View
aa75301b-5fff-4ad0-83cf-8ac4d5aa5f8a	SUCCESS	View
9bd83046-d1aa-4454-b27e-c203430e7e0c	SUCCESS	View

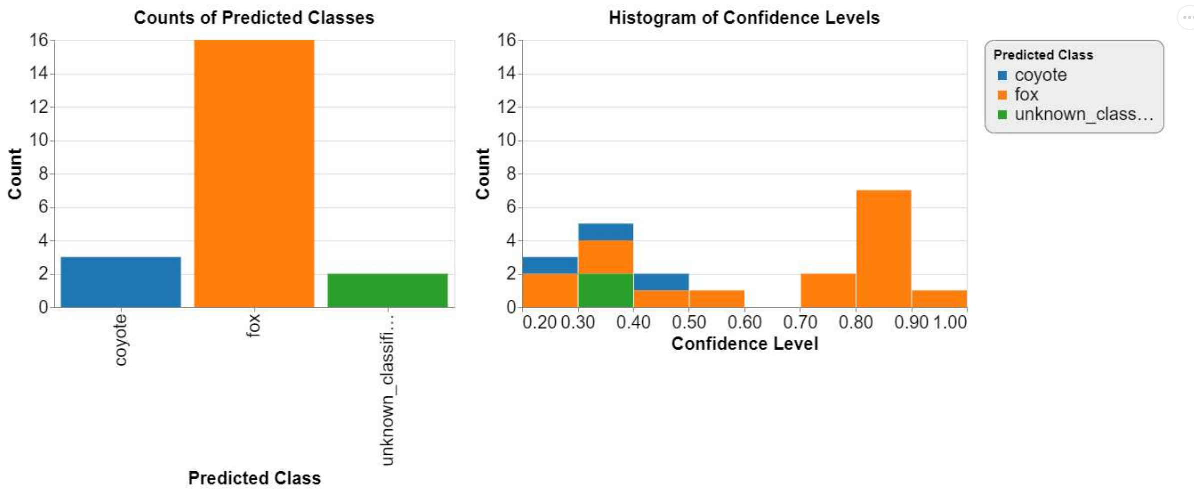
Analytics

Here we have our predictions of what animals (i.e. "model classes" or "classes" for short) were identified based on the model built from the custom labels you provided. On this webpage, we show some charts around what and when different classes were identified as well as allow you to change how sensitive the model predictions are before saving your final outputs.

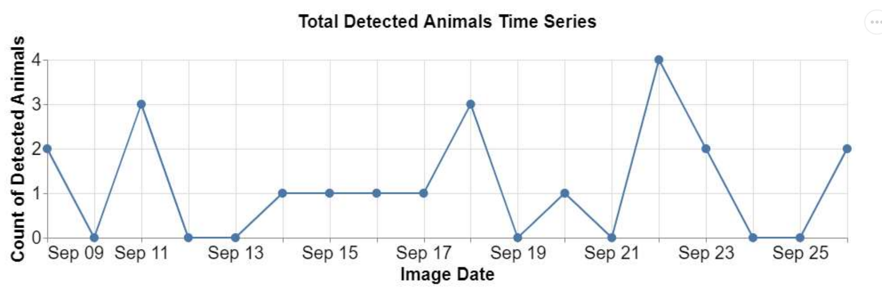
AniML - Classified Images Analytics and Review

Histograms of Model Class Predictions

Here are two histograms displaying the number of different classes identified (left) as well as the distribution of how confident the model was when classifying each image per class (i.e. the confidence scores per class). If a class has a low confidence score, more images during the training phase may be needed so that the model is better at recognizing the different angles, lighting exposures, etc. of that animal when trying to classify it. Changes to the model class cutoff scores in the next section will update these graphics. Click the three elipses '...' at the top right of each graphic if you wish to save them.



Next are two timeline charts that show how many animals were observed over time. Again, changes to the model class cutoff scores in the next section will update these graphics.



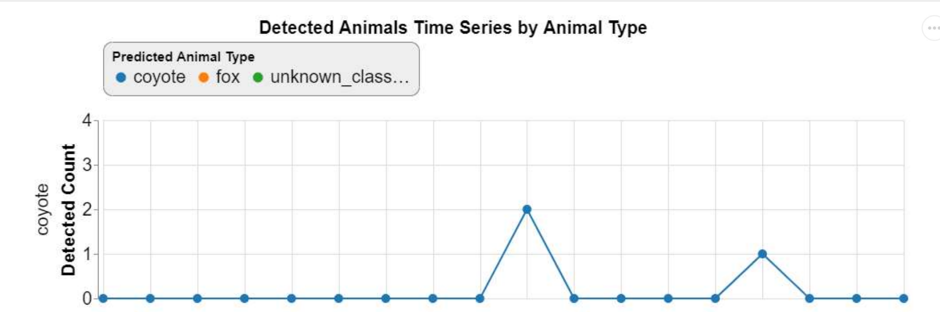


Image Classification Results

This table shows a random sample of images that the filtering model predicted as containing and not containing animals. To view more images in each class to get a sense of how the filterin model is performing, click the 'randomize images by class' button. You can adjust the sensitivity of how a model classifies an image as containing an animal or not in the next section below. When you are ready, you can save a zip file of your filtered results by clicking the 'Download zip' button at the top of either column.

Random Sample Images

Enter the animal type you wish to view and save the model classification results for.
To see a random sample of all the images, enter 'all_classes'.

e.g. fox or 'all_classes' [Randomize class images]

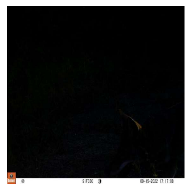
	count
fox	16
coyote	3
unknown_classification	2

09210100_JPG.rf.30eb39c49be1441a4f242a6165df2089.jpg

09240132_JPG.rf.f85fa1ad7e2fa899e82906fe3ec37b9e.jpg

09230122_JPG.rf.825237cd01557379e428eeac8963bfab.jpg

09110027_JPG.rf.e5b2d6597a99d0013498af7ddeff15e4.jpg



09150040_JPG.rf.ce44e70f5c299932d8eab5333b5cc0d2.jpg

09090019_JPG.rf.b8def9071a90f5525f050cb0e0d892f9.jpg

09260140_JPG.rf.92ae3de09b0a1e9f9353d7b07c814641.jpg

09250136_JPG.rf.53b58be7288a9844175406ed2680dd11.jpg

09220113_JPG.rf.67904d0764563fd6d921e907e034ab23.jpg

Adjust Image Classification Results

Like the filtering model, this classification model identifies a specific animal type as detected or not detected based on the confidence level that the image belongs to that class.

You may adjust the confidence thresholds for each class and see in the image results above how different confidence level thresholds affect the sensitivity of the model predictions above. You can then download the files associated with each animal class after any final adjustments to the confidence level cutoff.

The analytics histograms and timelines above also are updated along with any adjustments to the confidence level cutoffs in this section.

If you wish to view the inital model predictions, you can reset the table below.

Reclassify class with new confidence level

Keep class if confidence score greater than:

e.g. fox

e.g. 0.50

Submit

Confidence Score Cutoffs

class_name	confidence_cutoff
fox	default
coyote	default
unknown_classification	default

Any changes to predicted classes from new cutoff scores does not override any manual classifications below:

Manually reclassify individual images

If you see image(s) above that you want to manually reclassify without changing model confidence cutoff scores, you can manually reclassify them here.

Example format for single input:
img name = [image_name_1], new class = fox

Example format for multi input:
img name = [image_name_1, image_name_2], new class = fox

img name(s), e.g.: [image_name_1] or [img1, img2

new class, e.g.: fox

Submit

Reset classification

Reset the 'new' classifications to the default model outputs

Reset classifications

Initial Model Results

	image_ids	confidence_scores	predicted_classes_original
0	09220107_JPG.rf.1177492fb393be327df414b8180642db	0.84609	fox
1	09220112_JPG.rf.82197ddbc728a10e41d72941f8af6116	0.71150	fox
2	09220113_JPG.rf.67904d0764563fd6d921e907e034ab23	0.53751	fox
3	09220114_JPG.rf.d3671bfbdda727793746fdfbd12b7371	0.37580	fox
4	09200088_JPG.rf.f087b7a8a23bf745b932d1da2e4b2a93	0.31266	unknown_classification
5	09260140_JPG.rf.92ae3de09b0a1e9f9353d7b07c814641	0.92311	fox
6	09260141_JPG.rf.a95ea71af019915331d8918030a7ec9a	0.89160	fox
7	09230121_JPG.rf.be314214a792202918e4028ec5ec71e7	0.44693	coyote
8	09230121_JPG.rf.be314214a792202918e4028ec5ec71e7	0.22561	fox
9	09180079_JPG.rf.b655f8164e6875d1c2479aa844f757e8	0.70681	fox
10	09090019_JPG.rf.b8def9071a90f5525f050cb0e0d892f9	0.87569	fox
11	09090022_JPG.rf.ac8543bc7813616d7b3598404f6e3470	0.20554	fox
12	09110025_JPG.rf.9490847dd182c94e5cedb8fa5cf86500	0.89027	fox
13	09110027_JPG.rf.e5b2d6597a99d0013498af7ddeff15e4	0.80001	fox
14	09110028_JPG.rf.b89b5415b9eacf9342ecca992529923a	0.88212	fox
15	09140037_JPG.rf.b826ac1744e01dd10edf59d212e143b1	0.40213	fox
16	09150049_JPG.rf.f6700938d1c467deacb4458e30df8dfc	0.36607	unknown_classification
17	09160055_JPG.rf.07b029d123229fd2e960b688ca24eaac	0.80485	fox
18	09170068_JPG.rf.908e9552076f92367e02bacf0eeff9df	0.31640	fox
19	09180070_JPG.rf.da6b6b60b375f7e9a71fba931932cebd	0.35350	coyote
20	09180073_JPG.rf.7448f1e657b45ef54b66b72703577fa4	0.21910	coyote

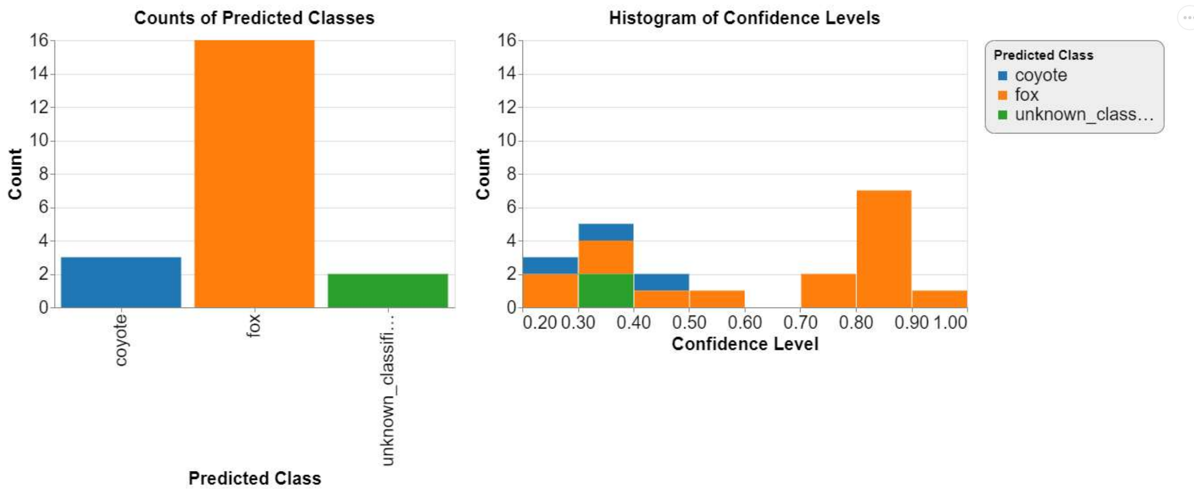
Analytics

Here we have our predictions of what animals (i.e. "model classes" or "classes" for short) were identified based on the model built from the custom labels you provided. On this webpage, we show some charts around what and when different classes were identified as well as allow you to change how sensitive the model predictions are before saving your final outputs.

AniML - Classified Images Analytics and Review

Histograms of Model Class Predictions

Here are two histograms displaying the number of different classes identified (left) as well as the distribution of how confident the model was when classifying each image per class (i.e. the confidence scores per class). If a class has a low confidence score, more images during the training phase may be needed so that the model is better at recognizing the different angles, lighting exposures, etc. of that animal when trying to classify it. Changes to the model class cutoff scores in the next section will update these graphics. Click the three elipses '...' at the top right of each graphic if you wish to save them.



Next are two timeline charts that show how many animals were observed over time. Again, changes to the model class cutoff scores in the next section will update these graphics.

Image Classification Results

This table shows a random sample of images that the filtering model predicted as containing and not containing animals. To view more images in each class to get a sense of how the filterin model is performing, click the 'randomize images by class' button. You can adjust the sensitivity of how a model classifies an image as containing an animal or not in the next section below. When you are ready, you can save a zip file of your filtered results by clicking the 'Download zip' button at the top of either column.

Random Sample Images

Enter the animal type you wish to view and save the model classification results for.
To see a random sample of all the images, enter 'all_classes'.

e.g. fox or 'all_classes' | Randomize class images

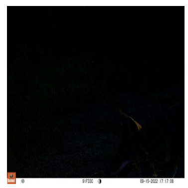
	count
fox	16
coyote	3
unknown_classification	2


09210100_JPG.rf.30eb39c49be1441a4f242a6165df2089.jpg


09240132_JPG.rf.f85fa1ad7e2fa899e82906fe3ec37b9e.jpg


09230122_JPG.rf.825237cd01557379e428eeac8963bfab.jpg

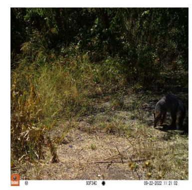

09110027_JPG.rf.e5b2d6597a99d0013498af7ddeff15e4.jpg


09150040_JPG.rf.ce44e70f5c299932d8eab5333b5cc0d2.jpg


09090019_JPG.rf.b8def9071a90f5525f050cb0e0d892f9.jpg


09260140_JPG.rf.92ae3de09b0a1e9f9353d7b07c814641.jpg


09250136_JPG.rf.53b58be7288a9844175406ed2680dd11.jpg


09220113_JPG.rf.67904d0764563fd6d921e907e034ab23.jpg

Adjust Image Classification Results

Like the filtering model, this classification model identifies a specific animal type as detected or not detected based on the confidence level that the image belongs to that class.

You may adjust the confidence thresholds for each class and see in the image results above how different confidence level thresholds affect the sensitivity of the model predictions above. You can then download the files associated with each animal class after any final adjustments to the confidence level cutoff.

The analytics histograms and timelines above also are updated along with any adjustments to the confidence level cutoffs in this section.

If you wish to view the inital model predictions, you can reset the table below.

Reclassify class with new confidence level

Keep class if confidence score greater than:

e.g. fox
e.g. 0.50
Submit

Confidence Score Cutoffs

class_name	confidence_cutoff
fox	default
coyote	default
unknown_classification	default

Any changes to predicted classes from new cutoff scores does not override any manual classifications below:

Manually reclassify individual images

If you see image(s) above that you want to manually reclassify without changing model confidence cutoff scores, you can manually reclassify them here.

Example format for single input:
img name = [image_name_1], new class = fox

Example format for multi input:
img name = [image_name_1, image_name_2], new class = fox

img name(s), e.g.: [image_name_1] or [img1, img2
new class, e.g.: fox
Submit

Reset classification

Reset the 'new' classifications to the default model outputs

Reset classifications

Initial Model Results

	image_ids	confidence_scores	predicted_classes_original
0	09220107_JPG.rf.1177492fb393be327df414b8180642db	0.84609	fox
1	09220112_JPG.rf.82197ddbc728a10e41d72941f8af6116	0.71150	fox
2	09220113_JPG.rf.67904d0764563fd6d921e907e034ab23	0.53751	fox
3	09220114_JPG.rf.d3671bfbdda727793746fdfbd12b7371	0.37580	fox
4	09200088_JPG.rf.f087b7a8a23bf745b932d1da2e4b2a93	0.31266	unknown_classification
5	09260140_JPG.rf.92ae3de09b0a1e9f9353d7b07c814641	0.92311	fox
6	09260141_JPG.rf.a95ea71af019915331d8918030a7ec9a	0.89160	fox
7	09230121_JPG.rf.be314214a792202918e4028ec5ec71e7	0.44693	coyote
8	09230121_JPG.rf.be314214a792202918e4028ec5ec71e7	0.22561	fox
9	09180079_JPG.rf.b655f8164e6875d1c2479aa844f757e8	0.70681	fox
10	09090019_JPG.rf.b8def9071a90f5525f050cb0e0d892f9	0.87569	fox
11	09090022_JPG.rf.ac8543bc7813616d7b3598404f6e3470	0.20554	fox
12	09110025_JPG.rf.9490847dd182c94e5cedb8fa5cf86500	0.89027	fox
13	09110027_JPG.rf.e5b2d6597a99d0013498af7ddeff15e4	0.80001	fox
14	09110028_JPG.rf.b89b5415b9eacf9342ecca992529923a	0.88212	fox
15	09140037_JPG.rf.b826ac1744e01dd10edf59d212e143b1	0.40213	fox
16	09150049_JPG.rf.f6700938d1c467deacb4458e30df8dfc	0.36607	unknown_classification
17	09160055_JPG.rf.07b029d123229fd2e960b688ca24eaac	0.80485	fox
18	09170068_JPG.rf.908e9552076f92367e02bacf0eeff9df	0.31640	fox
19	09180070_JPG.rf.da6b6b60b375f7e9a71fba931932cebd	0.35350	coyote
20	09180073_JPG.rf.7448f1e657b45ef54b66b72703577fa4	0.21910	coyote

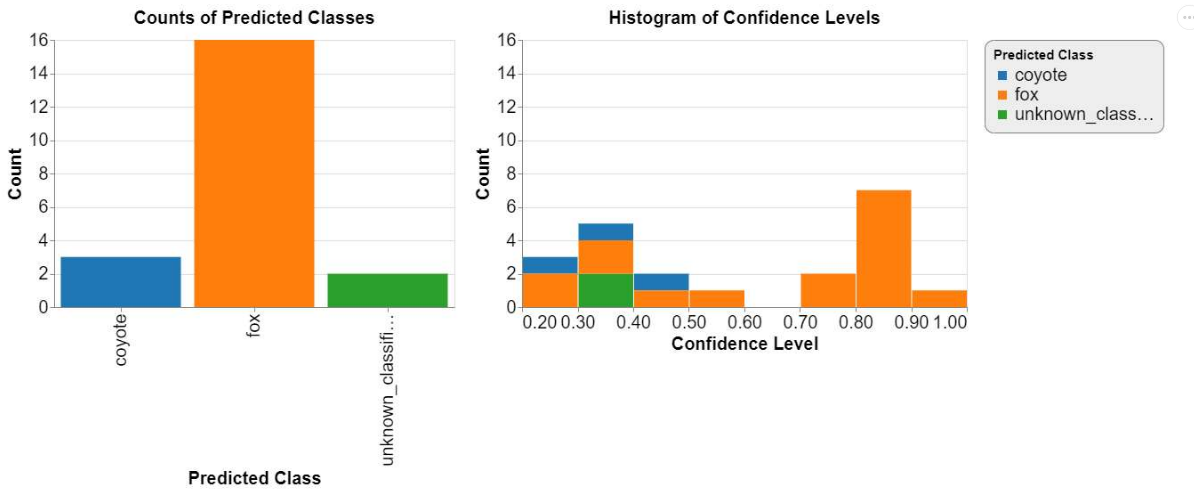
Analytics

Here we have our predictions of what animals (i.e. "model classes" or "classes" for short) were identified based on the model built from the custom labels you provided. On this webpage, we show some charts around what and when different classes were identified as well as allow you to change how sensitive the model predictions are before saving your final outputs.

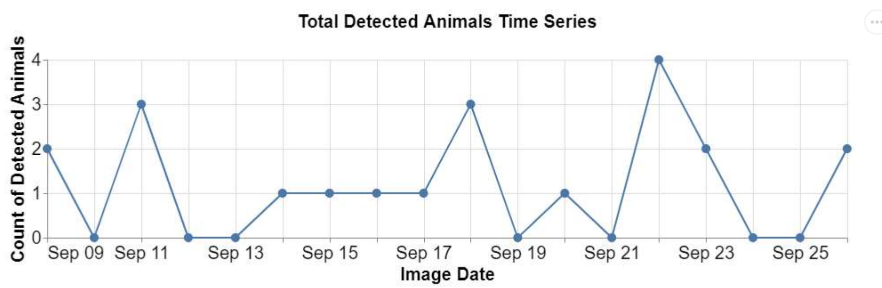
AniML - Classified Images Analytics and Review

Histograms of Model Class Predictions

Here are two histograms displaying the number of different classes identified (left) as well as the distribution of how confident the model was when classifying each image per class (i.e. the confidence scores per class). If a class has a low confidence score, more images during the training phase may be needed so that the model is better at recognizing the different angles, lighting exposures, etc. of that animal when trying to classify it. Changes to the model class cutoff scores in the next section will update these graphics. Click the three elipses '...' at the top right of each graphic if you wish to save them.



Next are two timeline charts that show how many animals were observed over time. Again, changes to the model class cutoff scores in the next section will update these graphics.



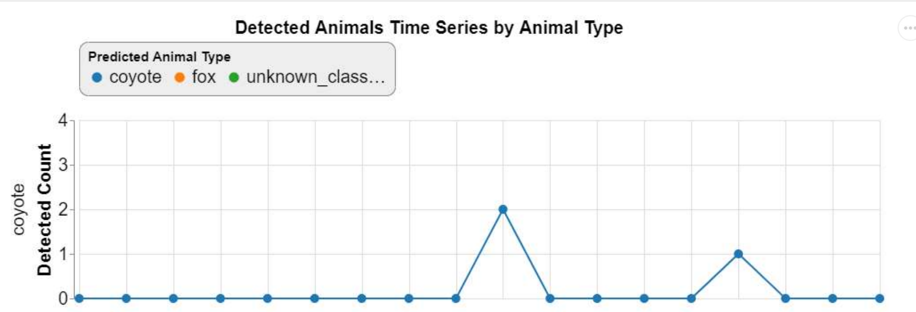


Image Classification Results

This table shows a random sample of images that the filtering model predicted as containing and not containing animals. To view more images in each class to get a sense of how the filterin model is performing, click the 'randomize images by class' button. You can adjust the sensitivity of how a model classifies an image as containing an animal or not in the next section below. When you are ready, you can save a zip file of your filtered results by clicking the 'Download zip' button at the top of either column.

Random Sample Images

Enter the animal type you wish to view and save the model classification results for.
To see a random sample of all the images, enter 'all_classes'.

e.g. fox or 'all_classes' Randomize class images

	count
fox	16
coyote	3
unknown_classification	2

Model Predictions

Low confidence that image contains: fox

Download zip of prediction of not_fox images

Class_not_fox
not enough images in class

Class_not_fox
not enough images in class

Class_not_fox
not enough images in class

Class_not_fox
not enough images in class

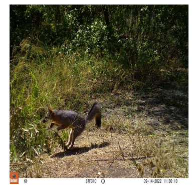
Class_not_fox
not enough images in class

Class_not_fox
not enough images in class

Model Predictions

High confidence that image contains: fox

Download zip of prediction of fox images



09140037_JPG.rf.b826ac1744e01dd10edf59d212e143b1.jpg

09090019_JPG.rf.b8def9071a90f5525f050cb0e0d892f9.jpg

09110027_JPG.rf.e5b2d6597a99d0013498af7ddeff15e4.jpg

09260141_JPG.rf.a95ea71af019915331d8918030a7ec9a.jpg

09110028_JPG.rf.b89b5415b9eacf9342ecca992529923a.jpg

09220107_JPG.rf.1177492fb393be327df414b8180642db.jpg

Class_not_fox
not enough images in class

09220114_JPG.rf.d3671bfbdda727793746fdfbd12b7371.jpg

Class_not_fox
not enough images in class

09090022_JPG.rf.ac8543bc7813616d7b3598404f6e3470.jpg

Adjust Image Classification Results

Like the filtering model, this classification model identifies a specific animal type as detected or not detected based on the confidence level that the image belongs to that class.

You may adjust the confidence thresholds for each class and see in the image results above how different confidence level thresholds affect the sensitivity of the model predictions above. You can then download the files associated with each animal class after any final adjustments to the confidence level cutoff.

The analytics histograms and timelines above also are updated along with any adjustments to the confidence level cutoffs in this section.

If you wish to view the inital model predictions, you can reset the table below.

Reclassify class with new confidence level

Keep class if confidence score greater than:

e.g. fox
e.g. 0.50
Submit

Confidence Score Cutoffs

class_name	confidence_cutoff
fox	default
coyote	default
unknown_classification	default

Any changes to predicted classes from new cutoff scores does not override any manual classifications below:

Manually reclassify individual images

If you see image(s) above that you want to manually reclassify without changing model confidence cutoff scores, you can manually reclassify them here.

Example format for single input:
img name = [image_name_1], new class = fox

Example format for multi input:
img name = [image_name_1, image_name_2], new class = fox

img name(s), e.g.: [image_name_1] or [img1, img2
new class, e.g.: fox
Submit

Reset classification

Reset the 'new' classifications to the default model outputs

Reset classifications

Initial Model Results

	image_ids	confidence_scores	predicted_classes_original
0	09220107_JPG.rf.1177492fb393be327df414b8180642db	0.84609	fox
1	09220112_JPG.rf.82197ddbc728a10e41d72941f8af6116	0.71150	fox
2	09220113_JPG.rf.67904d0764563fd6d921e907e034ab23	0.53751	fox
3	09220114_JPG.rf.d3671bfbdda727793746fdfbd12b7371	0.37580	fox
4	09200088_JPG.rf.f087b7a8a23bf745b932d1da2e4b2a93	0.31266	unknown_classification
5	09260140_JPG.rf.92ae3de09b0a1e9f9353d7b07c814641	0.92311	fox
6	09260141_JPG.rf.a95ea71af019915331d8918030a7ec9a	0.89160	fox
7	09230121_JPG.rf.be314214a792202918e4028ec5ec71e7	0.44693	coyote
8	09230121_JPG.rf.be314214a792202918e4028ec5ec71e7	0.22561	fox
9	09180079_JPG.rf.b655f8164e6875d1c2479aa844f757e8	0.70681	fox
10	09090019_JPG.rf.b8def9071a90f5525f050cb0e0d892f9	0.87569	fox
11	09090022_JPG.rf.ac8543bc7813616d7b3598404f6e3470	0.20554	fox
12	09110025_JPG.rf.9490847dd182c94e5cedb8fa5cf86500	0.89027	fox
13	09110027_JPG.rf.e5b2d6597a99d0013498af7ddeff15e4	0.80001	fox
14	09110028_JPG.rf.b89b5415b9eacf9342ecca992529923a	0.88212	fox
15	09140037_JPG.rf.b826ac1744e01dd10edf59d212e143b1	0.40213	fox
16	09150049_JPG.rf.f6700938d1c467deacb4458e30df8dfc	0.36607	unknown_classification
17	09160055_JPG.rf.07b029d123229fd2e960b688ca24eaac	0.80485	fox
18	09170068_JPG.rf.908e9552076f92367e02bacf0eeff9df	0.31640	fox
19	09180070_JPG.rf.da6b6b60b375f7e9a71fba931932cebd	0.35350	coyote
20	09180073_JPG.rf.7448f1e657b45ef54b66b72703577fa4	0.21910	coyote

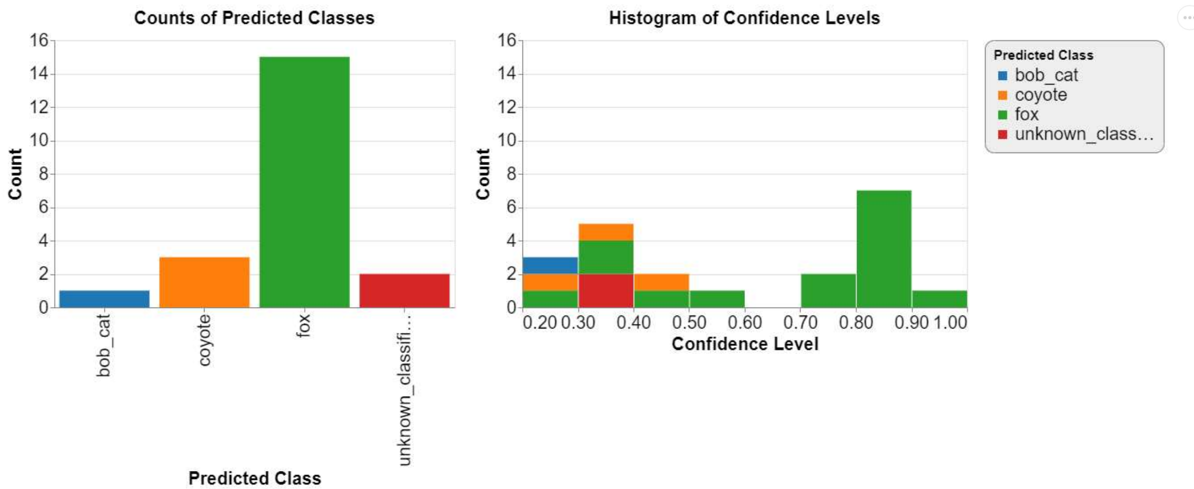
Analytics

Here we have our predictions of what animals (i.e. "model classes" or "classes" for short) were identified based on the model built from the custom labels you provided. On this webpage, we show some charts around what and when different classes were identified as well as allow you to change how sensitive the model predictions are before saving your final outputs.

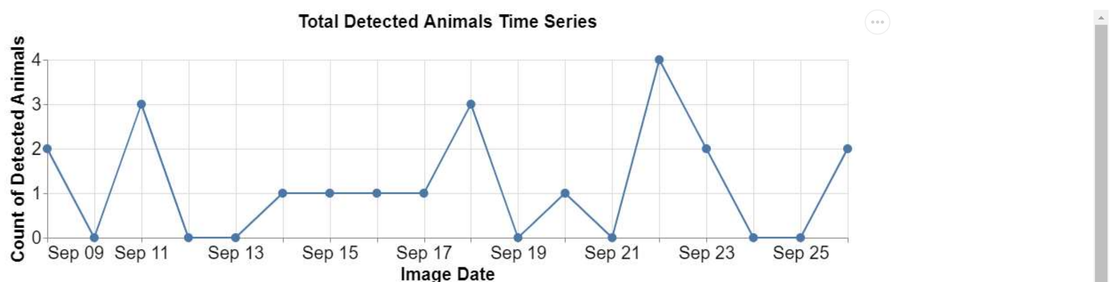
AniML - Classified Images Analytics and Review

Histograms of Model Class Predictions

Here are two histograms displaying the number of different classes identified (left) as well as the distribution of how confident the model was when classifying each image per class (i.e. the confidence scores per class). If a class has a low confidence score, more images during the training phase may be needed so that the model is better at recognizing the different angles, lighting exposures, etc. of that animal when trying to classify it. Changes to the model class cutoff scores in the next section will update these graphics. Click the three elipses '...' at the top right of each graphic if you wish to save them.



Next are two timeline charts that show how many animals were observed over time. Again, changes to the model class cutoff scores in the next section will update these graphics.



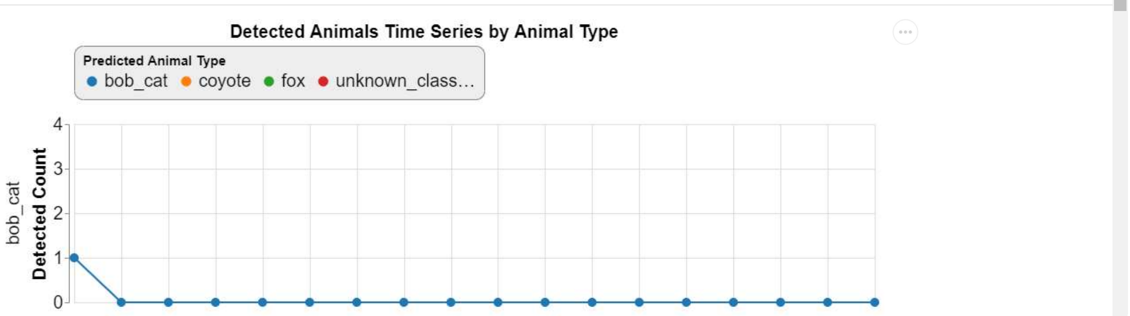


Image Classification Results

This table shows a random sample of images that the filtering model predicted as containing and not containing animals. To view more images in each class to get a sense of how the filterin model is performing, click the 'randomize images by class' button. You can adjust the sensitivity of how a model classifies an image as containing an animal or not in the next section below. When you are ready, you can save a zip file of your filtered results by clicking the 'Download zip' button at the top of either column.

Random Sample Images

Enter the animal type you wish to view and save the model classification results for.
To see a random sample of all the images, enter 'all_classes'.

e.g. fox or 'all_classes' | Randomize class images

predicted_classes_original	predicted_classes_new	count
coyote	coyote	3
fox	bob_cat	1
fox	fox	15
unknown_classification	unknown_classification	2

Model Predictions

Low confidence that image contains: bob_cat

Download zip of prediction of not_bob_cat images

Class_not_bob_cat
not enough images in class

Class_not_bob_cat
not enough images in class

Class_not_bob_cat
not enough images in class

Class_not_bob_cat
not enough images in class

Class_not_bob_cat
not enough images in class

Class_not_bob_cat
not enough images in class

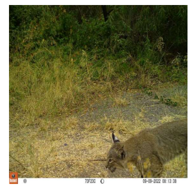
Class_not_bob_cat
not enough images in class

Class_not_bob_cat
not enough images in class

Model Predictions

High confidence that image contains: bob_cat

Download zip of prediction of bob_cat images



09090022_JPG.rf.ac8543bc7813616d7b3598404f6e3470.jpg

Class_bob_cat
not enough images in class

Class_bob_cat
not enough images in class

Class_bob_cat
not enough images in class

Class_bob_cat
not enough images in class

Class_bob_cat
not enough images in class

Class_bob_cat
not enough images in class

Class_bob_cat
not enough images in class

Adjust Image Classification Results

Like the filtering model, this classification model identifies a specific animal type as detected or not detected based on the confidence level that the image belongs to that class.

You may adjust the confidence thresholds for each class and see in the image results above how different confidence level thresholds affect the sensitivity of the model predictions above. You can then download the files associated with each animal class after any final adjustments to the confidence level cutoff.

The analytics histograms and timelines above also are updated along with any adjustments to the confidence level cutoffs in this section.

If you wish to view the inital model predictions, you can reset the table below.

Reclassify class with new confidence level

Keep class if confidence score greater than:

e.g. fox

e.g. 0.50

Submit

Confidence Score Cutoffs

class_name	confidence_cutoff
fox	default
coyote	default
unknown_classification	default

Any changes to predicted classes from new cutoff scores does not override any manual classifications below:

Manually reclassify individual images

If you see image(s) above that you want to manually reclassify without changing model confidence cutoff scores, you can manually reclassify them here.

Example format for single input:
img name = [image_name_1], new class = fox

Example format for multi input:
img name = [image_name_1, image_name_2], new class = fox

img name(s), e.g.: [image_name_1] or [img1, img2

new class, e.g.: fox

Submit

Reset classification

Reset the 'new' classifications to the default model outputs

Reset classifications

Adjusted Model Results

	image_ids	confidence_scores	predicted_classes_original	predicted_classes_new	manually_reclassified
0	09220107_JPG.rf.1177492fb393be327df414b8180642db	0.84609	fox	fox	no
1	09220112_JPG.rf.82197ddbc728a10e41d72941f8af6116	0.71150	fox	fox	no
2	09220113_JPG.rf.67904d0764563fd6d921e907e034ab23	0.53751	fox	fox	no
3	09220114_JPG.rf.d3671bfbdda727793746fdfbd12b7371	0.37580	fox	fox	no
4	09200088_JPG.rf.f087b7a8a23bf745b932d1da2e4b2a93	0.31266	unknown_classification	unknown_classification	no
5	09260140_JPG.rf.92ae3de09b0a1e9f9353d7b07c814641	0.92311	fox	fox	no
6	09260141_JPG.rf.a95ea71af019915331d8918030a7ec9a	0.89160	fox	fox	no
7	09230121_JPG.rf.be314214a792202918e4028ec5ec71e7	0.44693	coyote	coyote	no
8	09230121_JPG.rf.be314214a792202918e4028ec5ec71e7	0.22561	fox	fox	no
9	09180079_JPG.rf.b655f8164e6875d1c2479aa844f757e8	0.70681	fox	fox	no
10	09090019_JPG.rf.b8def9071a90f5525f050cb0e0d892f9	0.87569	fox	fox	no
11	09090022_JPG.rf.ac8543bc7813616d7b3598404f6e3470	0.20554	fox	bob_cat	yes
12	09110025_JPG.rf.9490847dd182c94e5cedb8fa5cf86500	0.89027	fox	fox	no
13	09110027_JPG.rf.e5b2d6597a99d0013498af7ddeff15e4	0.80001	fox	fox	no
14	09110028_JPG.rf.b89b5415b9eacf9342ecca992529923a	0.88212	fox	fox	no
15	09140037_JPG.rf.b826ac1744e01dd10edf59d212e143b1	0.40213	fox	fox	no
16	09150049_JPG.rf.f6700938d1c467deacb4458e30df8dfc	0.36607	unknown_classification	unknown_classification	no
17	09160055_JPG.rf.07b029d123229fd2e960b688ca24eaac	0.80485	fox	fox	no
18	09170068_JPG.rf.908e9552076f92367e02bacf0eeff9df	0.31640	fox	fox	no
19	09180070_JPG.rf.da6b6b60b375f7e9a71fba931932cebd	0.35350	coyote	coyote	no
20	09180073_JPG.rf.7448f1e657b45ef54b66b72703577fa4	0.21910	coyote	coyote	no

About

AniML - About

Specialists with little to no computer vision experience historically have not had access to AI tools to better improve their manual image processing tasks or increase their processing throughput.

These users need a way to rapidly create a computer vision (CV) system to:

1. Automatically filter a large set of images,
2. Identify objects of interest, and
3. Provide analytical insights

thus reducing the amount of time needed for non-technical tasks enables these groups to spend more time where it counts.

With the AniML tool, different industries can unlock the power of CV, such as:

- Biologist will be better equiped to quickly identifies species images
- Life scientists and medical professional can capture and analyze microscopic cell imagery or medical imagery like x-rays (link)
- QA Manufacturer can better detect equipment defects (link)

AI Information and Research Background

AniML - Model Background

Model Decision and Background:

The network architecture of Yolov5 , the algorithm used for the AniML AI model engine, consists of three parts:

1. Backbone: CSPNet or CSPDarknet
2. Neck: PANet, and
3. Head: Yolo Layer.

The data are first input to CSPDarknet for feature extraction, and then fed to PANet for feature fusion. Finally, Yolov5 Layer outputs detection results, including the predicted class, confidence score for that class prediction, the bounding box location and size which encompassing the object of interest.

The scheme of the YOLOv5 Architecture as Convolutional Neural Network (CNN), in which the key parts are the BackBone, Neck and Head.

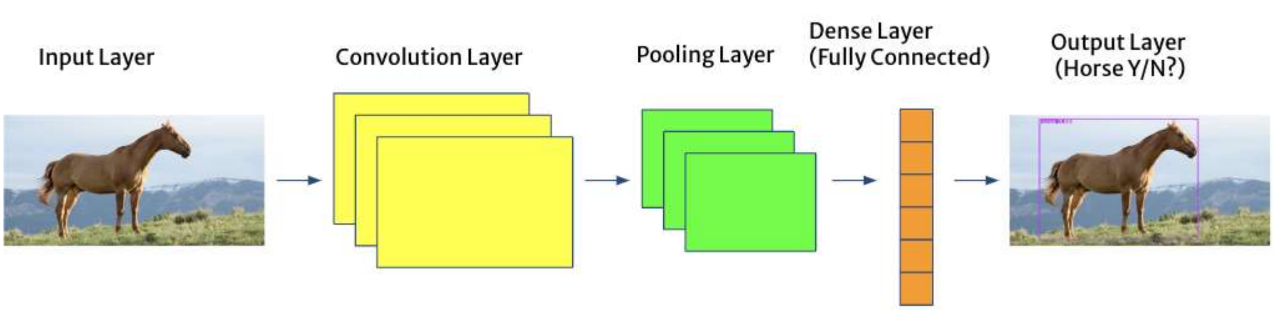
- BackBone CSPNet is used in order to extract features from the images which are used as input images.
- Neck is used for the creation of pyramid feature. It helps the module on scaling factor of detected objects which are of the same nature but different scales. The technique which is used for creation of pyramid features is PANet.
- Head is to apply anchor of different sizes on those features which are generated in the previous layers with value of probability as well as bounding box with score.

Mask R-CNN is a state of the art model for instance segmentation, developed on top of Faster R-CNN. Faster R-CNN is a region-based convolutional neural networks [2], that returns bounding boxes for each object and its class label with a confidence score.

The default YoloV5 model was trained Common Object in Context (COCO) dataset, which contains 80 different classes. AniML seeks to retrain the final layers of YoloV5 with some additional layers that are customized to an end users specific use case.

Convolutional Neural Network

Decision Layers for Computer Vision Object Detection



Further Background Reading

CNN Background:

CNN is a kind of Network architecture for deep-learning algorithms for identifying and recognizing objects (pixel data process). CNN leverages principles from linear algebra, such as matrix multiplication to identify patterns within an image and is highly suitable for image classification and CV applications with highly accurate results, especially when a lot of data is involved. i.e. self-driving cars and facial recognition.

CNN Layers:

The CNN learns the object's features in successive iterations as the object data moves through the CNN's many layers. This direct (and deep) learning eliminates the need for manual feature extraction (feature engineering).

A CNN can have multiple layers, each of layer learns to detect the different features of an input image. A filter or kernel is applied to each image to produce an output that gets progressively better and more detailed after each layer. As each filter activates certain features from the image, it does its work and passes on its output to the filter in the next layer.

Each layer learns to identify different features and the operations end up being repeated for dozens, hundreds or even thousands of layers.

Finally, all the image data progressing through the CNN's multiple layers allow the CNN to identify the entire object.

CNN Applications:

CNNs can be retrained for new recognition tasks and built on pre-existing networks and can be used for real-world applications without increasing computational complexities or costs.

Other CV Modeling Techniques:

One type of an Artificial neural networks (ANNs) is a recurrent neural network (RNN) that uses sequential or time series data as input. It is suitable for applications involving NLP, language translation, speech recognition and image captioning.

Contact Information

If you have any questions or suggestions, do not hesitate to reach out!

AniML - Contact

Your name:

Your email:

Your message:

submit form

Response

Contact Information

If you have any questions or suggestions, do not hesitate to reach out!

AniML - Contact

Your name: Erin

Your email: example@email.com

Your message:

You saved me so much time!

submit form

Response

"successfully received your message!"