SCC and IndySCC

SC24

Atlanta, GA | hpc creates.

Mystery Application

FIND MY CAT

HHC

SC24 Atlanta, GA | hpc creates.

Mystery Application

Application Creative Concept and Visual Design by

Hector H. Corzo (ORNL) and Jens Glaser (ORNL)

November 2024

Find My Cat Challenge

Challenge Overview

Welcome to the Find My Cat Challenge. Motivated by the essential goal of locating our beloved felines when they hide and we can't find them, the goal of this challenge is to develop an innovative, solution for detecting hidden cats in diverse and cluttered environments.

Why This Challenge Matters

While finding a cat might seem like a playful pursuit (which is part of the goal for this challenge), object detection is a fundamental aspect of modern computer vision. Our objective is to evaluate your creativity and adaptability in designing practical solutions for real-world challenges with real-world constraints, such as energy efficiency and solution reproducibility. These factors are crucial for deploying computer vision systems in resource-limited and mission-critical environments. The core technologies and techniques you will develop in this challenge are directly applicable to more serious fields such as:

- **Wildlife conservation:** Monitor endangered species in their natural habitats.
- **Search & rescue operations:** Help locate missing persons amidst debris or dense foliage.
- **Autonomous vehicles:** Improve obstacle detection for safer self-driving cars.
- **Medical imaging:** Enhance disease diagnosis and treatment through accurate image processing.

Challenge Details

The primary objective is to develop a computational application to identify cats within images. Your developed solution must be submitted as an Jupyter notebook compatible with Google Colab for testing and verification. To ensure accessibility and fairness for all participants, we encourage you to design and develop your solution with the end goal of efficient performance on moderate resources, similar to those in a standard **Jupyter notebook environment with 16 CPUs, 16GB of memory, and a single V100 or T4 NVIDIA GPU.** You are free to use any computational tools, algorithms, or models you find suitable, provided that your solution is self-contained and adheres to the specified submission requirements for evaluation.

Provided Data

Participants will be provided with 60 annotated images featuring cats in a variety of complex and diverse settings. These images are categorized as:

- **Easy:** 30 images
- **Medium:** 20 images
- **Hard:** 10 images

These images can be used as training samples, validation sets, or leveraged in any manner participants deem effective for developing their solutions.

Evaluation Data

Submitted solutions will be evaluated on 400 unseen images, divided as follows:

- **Easy:** 200 images
- **Medium:** 100 images
- **Hard:** 100 images

Both the provided and evaluation datasets will be used for the evaluation of the effectiveness of the solution.

Submission Requirements

Google Colab Compatibility

The notebook should meet the following criteria:

1. **Google Colab:** Ensure that the solution is fully compatible with Google Colab, if additional libraries are needed please provide them.
2. **Flexible Directory Selection:** Enable easy modification of the directory path so users can switch between different image sets. Changing the directory name should instantly allow testing with new image data.
3. **Result Display:** Ensure the notebook displays detection results on images as specified in the evaluation guidelines, providing clear, visual outputs for easy assessment.

Documentation

Submissions should include a well-structured document (Notebook, Markdown, etc.) that covers the following:

a. **Clear Instructions:** Provide detailed, step-by-step instructions and any necessary scripts to ensure seamless execution of the your solution.
b. **Development Process:** Outline the methodology and processes followed during the development phase, and mention any additional datasets used.
c. **Solution Rationale:** Describe the logic behind your approach, including relevant algorithms, techniques, and frameworks.
d. **Justification of Approach:** Make a strong case for why your solution is the most effective, highlighting its strengths and advantages over alternatives.
e. **Resource Usage and Optimization:** Estimate the compute resources, energy consumption, GPU utilization, training time, and CO_2 emissions. Highlight any optimizations or trade-offs made to reduce resource usage.
f. **Iterative Improvements:** Discuss any challenges encountered and the adjustments made to refine your methodology based on initial findings.

Evaluation Metrics

While you may use more powerful local hardware during development, please ensure your final submission performs smoothly within the resources limits of a standard **Jupyter notebook environment with 16 CPUs, 16GB of memory, and a single V100 GPU.** While exceptions may be considered on a case-by-case basis, they are not guaranteed. Therefore, we recommend designing your solution with these resource constraints in mind to optimize both performance and compatibility.

Deliverables

1. Executable Notebook:

- Must be a standalone Google Colab notebook where directories can be modified easily to test different image sets.
- The first test set will be the 60 provided images, with correct detection required in at least 30 images.
- The second test set will consist of 400 unseen images, which will be evaluated over ten consecutive runs.

2. Documentation:

- Include detailed instructions for the execution of your notebook and explanations covering the development process of your solution.

Testing and Scoring Criteria

1. Cat Detection Performance

- Teams that correctly detect at least 40% of images across all difficulty levels to qualify for 300 Points.
- Scoring by image detected.
  - **Easy :** 2 points per image detected **(200 images) – Max 400 points.**
  - **Medium :** 5 points per image detected **(100 images) – Max 500 points.**
  - **Hard :** 10 points per image detected **(100 images) – Max 1000 points.**

2. Precision and Recall

- The evaluation dataset will be run 10 times. Consistent detection across these runs (detecting the same images each time) will earn **150 points.**
- Precision (**50 points**):

$$\frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

- Recall (**50 points**):

$$\frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

3. Energy Efficiency

- Energy Consumption per Detection during Inference: The team with the best energy efficiency per detection will receive an additional **200 points**, the second place will receive **150 points**, and the third place will receive **100 points**. The energy efficiency in this challenge will be computed as:

$$\frac{\text{Total Energy Consumption (J)}}{\text{Number of Cats Detected}}$$

- If a team achieves a significantly better energy efficiency score than the average, they will receive an additional **100 points**.

4. Documentation and Transparency

- Teams that submit their solution notebook, along with clear documentation detailing the development process, energy considerations for testing, and any computational insights involved in developing their solutions, will receive **800 points** according to the following criteria:
  - **Full Solution:** The solution notebook is submitted as requested (**500 points**).
  - **Partial Solution:** Only part of the solution notebook is submitted (**350 points**).
  - **Full documentation:** The documentation is clear, well structured and complete (**300 points**).
  - **Partial Documentation:** Only part of the solution notebook is submitted (**100 points**).

Bonus Points

1. **Additional Documentation:** Up to **250 points** will be awarded for submitting documentation with clear metrics on energy consumption during solution development, including both actual energy used in the current submission and estimates for full reproduction with lessons learned.
2. **Detailed Reporting:** The team with the most comprehensive documentation, covering solution architecture, development processes, hyperparameters, evaluation metrics, hardware considerations, and key insights, will receive **200 points.**
3. **Design Insights:** Up to **100 points** will be given for insights on potential improvements (e.g., algorithm, hardware).
4. **Detection Bonus:**
   - **60% Failure Rate:** If your solution detects an image missed by 60% of other teams, scoring is as follows:
     - **Easy:** 2.25 points
     - **Medium:** 5.5 points
     - **Hard:** 12 points
   - **80% Failure Rate:** If your solution detects an image missed by 80% of other teams, scoring is as follows:
     - **Easy:** 3 points
     - **Medium:** 7 points
     - **Hard:** 15 points
   - **Perfect Detection:** If your solution consistently detects all datasets with 100% accuracy, you will receive an additional 10% of the total bonus detection points.
   - **Accuracy Threshold:** Teams achieving a detection accuracy of 70% or higher while maintaining energy efficiency (energy consumption below the median) will receive an additional **100 points**. If the detection accuracy reaches 81% or higher, the team will receive the initial bonus plus an additional **150 points.**

3. Precision and Recall:

- **Reproducibility (200 Points):** Awarded to the team with the highest consistency across 10 independent runs; the team with the most consistent precision and recall across the 10 runs.
- **Robustness (200 Points):** Awarded to the team with the best performance consistency across the easy, medium, and hard difficulty levels. This means your solution maintains stable and reliable detection performance regardless of the complexity or challenge of the images they are processing.

Penalties

1. **Resource Limitations:** Solutions exceeding the specified computational capacity for inference incurs a **35% penalty** on the highest-performing score and an additional **10% reduction** from the total score.
2. **Precision-Recall:** Of the four lowest-performing teams in precision-recall, the team with the lowest score faces a **10% reduction** on their highest-scoring section, while the other three incur a **5% reduction.**
3. **Energy Consumption:** Among the four highest energy-consuming teams, the top energy consumer faces a **10% reduction** on their highest-scoring section, with the remaining three facing a **5% reduction** on their total score.
4. **Solution Reproducibility and Robustness:** The team with the lowest score in this category faces a **10% reduction** on their highest-scoring section, while the other three face a **5% reduction.**

Optional Challenge

You have the opportunity to design a unique picture test aimed to challenge other team's solutions while showcasing the strengths of your own. Your picture must include a cat (or cats) and should resemble a realistic scene that could be captured in real life—no overly unrealistic or fantastical scenarios. Other elements in the picture can also be realistic items commonly found in real life. If your team chooses to participate, the submission requirements are as follows:

1. **Image Specifications**
   - **Inclusion of Cats:** The image must include at least one cat.
   - **Realistic Themes:** The theme should be based on realistic, real-life scenarios (e.g., everyday environments).
   - **Realistic Elements:** Other elements in the image must also be realistically attainable.
2. **Explanation Document**
   - **Idea Development:** Describe how you conceptualized the picture. Explain any observations during the development of your solution that highlighted weaknesses you could exploit.
   - **Technical Justification:** Provide a technical explanation of why your solution can detect the cat(s) in your image while others may not.
3. **Validation**
   - **Evidence of Detection:** Include evidence demonstrating that your solution successfully detects the cat(s) in your submitted image.

Scoring

1. **If You Submit a Picture but not all the teams do**

   Points Allocation:

   - **Initial Submission:** 20 points if at least 10% of other teams' solutions fail to detect the cat.
   - **Intermediate Submission:** 30 additional points if 50% of teams cannot detect the cat.
   - **Advanced Submission:** 40 additional points if 70% of teams cannot detect the cat.
   - **Cumulative Points:** Points accumulate based on the percentage thresholds met, with a maximum of 90 points per team submission.

2. **If All Teams Submit a Picture**

   For each picture challenge:

   - **Successful Detection by Your Solution:** +60 points per successful pass.
   - **Failure by Other Teams' Solutions to Detect Your Picture:** +40 points per failure induced.
   - **Failure to Detect Other Teams' Pictures:** -30 points per failure.

   Bonuses and Penalties:

   - **High Success Rate Bonus (HSR):** +150 points if your solution successfully detects more than 75% of challenges.
   - **High Failure Rate Penalty (HFR):** -150 points if your solution fails to identify more than 50% of challenges.
   - **Most Successful Picture Detection:** +150 points if your solution detects the highest number of pictures.

Example Scenarios

Before submitting a picture for the optional challenge, consider how it may impact your score for the first part of the challenge.

Partial Picture Submissions:

- **Total Teams**: 11
- **Picture Submissions:** 8 teams submitted a challenging picture.
- **Maximum Points:**
  - Induced Failures: 7 teams fail to identify your cat (20 + 30 + 40 = 90 points).
  - Total: **90 points**.

All Teams Submit Pictures:

- **Total Teams:** 11
- **Picture Submissions:** 11 teams submitted a challenging picture.
- **Maximum Points:**
  - Induced Failures: 9 teams fail to identify your cat (9 × 40 = 360 points).
  - Successful Detections: Your solution identifies 9 out of the 10 submitted pictures (9 × 60 = 540 points).
  - Detection Failure: Your solution failed to identify the cat in one of the 10 submitted pictures (1 × −30 = −30 points).
  - HSR Bonus: 150 points.
  - Best Detection Bonus: 150 points for identifying the most pictures.
  - Total Possible Points: Up to **1170 points**.

Before Submitting

Ensure full requirements compliance before submitting:

1. **Setup Instructions:** A clear, step-by-step guide for setting up the environment and running the code.
2. **Software Versions:** List all software versions used (e.g., Python, libraries).
3. **Dependencies:** Specify all required libraries and their versions.
4. **Tools Used:** Detail additional tools used in the solution.
5. **Usage Instructions:** Explain how to execute your solution, including command-line arguments or configurations.
6. **Code Structure:** Provide an overview of your code structure and key components.
7. **Pre-trained Model Files:** Verify inclusion of any necessary pre-trained models.
8. **Hardware Verification:** Confirm solution compatibility with specified hardware.
9. **Detailed Logs:** Include clear, well-formatted logs of energy consumption during training and inference.
10. **Performance Reports:** Verify detection performance metrics using the provided dataset.

We understand the time constraints, so just do your best with the deliverables. Our main focus is on the creativity and effectiveness of your solution, rather than its computational complexity or the level of detail in the final deliverables. While they don't need to be perfect, they should be sufficient to allow us to properly evaluate your solution.

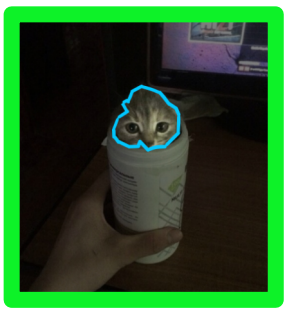
Solution Specifications

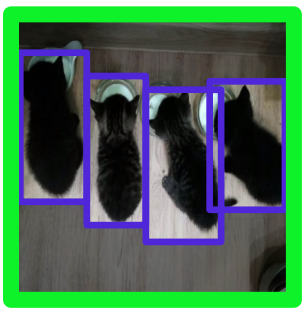
- The objective of this application is to create a computational tool that accurately identifies cats in various images. A successful identification will be achieved if your application detects a cat by either outlining its silhouette or placing a clear bounding box around it.
    - **Accuracy Requirement:** Your application must avoid selecting non-cat objects in the image. Any selection that includes non-cat elements will be marked as incorrect, as shown in the example provided.
    - **Camouflage and Complex Backgrounds:** In images where the cat is partially camouflaged or situated on or among other objects (such as towels or other animals), if your solution includes parts of these objects as part of the cat detection, the identification will be deemed invalid, as illustrated in the example below.
    - **Multiple Cats:** In images with more than one cat, your solution must identify all cats in the image to be considered correct. For example, if there are multiple cats present, each cat must be detected individually.
    - **Collage Images:** For medium and hard levels, test images may be collages made by combining multiple individual images from the previous level. In these cases, all cats in every section of the collage must be detected for the solution to be marked correct.
- Design your solution to output the following:
    1. Each image with an annotation marking where the cat is located.
    2. A summary at the end of the execution, after all images have been processed and annotated, listing:
        - The name of each image.
        - The number of cats detected in each image.
        - If no cats are detected, the image name should indicate that no cats were found.

Detection Examples

Valid Detections

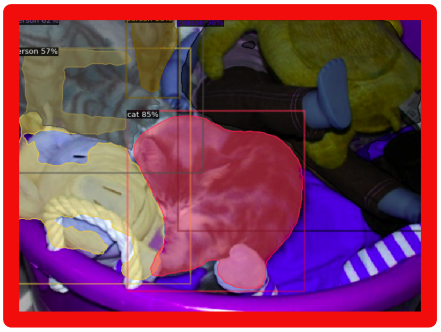


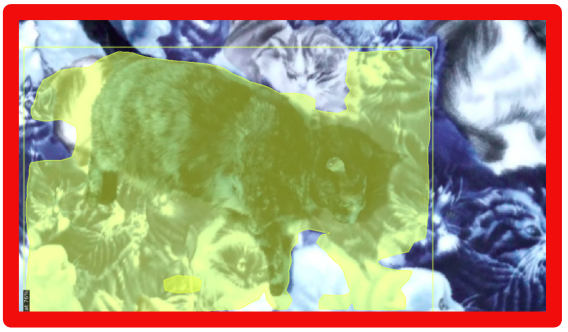


Invalid Detections







Additional Information

- Starting a jupyter lab instance on a local server and connecting to it via SSH tunnel.

```
$ ssh -L 8888:localhost:8888 myusername@myserver.com
```

Once logged in, install miniconda3 (https://docs.anaconda.com/miniconda/).

```
$ conda install -q jupyterlab
$ jupyter lab
```

In a web browser on the machine you launched the ssh command, open the URL https://localhost:8888 , which opens the jupyter lab console.

- For submission of your solution, save the notebook as a ".ipynb" file and confirm that it can be uploaded to and that it runs on colab.google.com (with all necessary weights and software dependencies) using T4 GPUs (Runtime -> Change Runtime Type -> Hardware accelerator -> T4 GPU).

- Repositories for energy measurements:
  - Zeus: https://github.com/ml-energy/zeus
  - CodeCarbon: (https://github.com/mlco2/codecarbon
  - Impact: https://github.com/mlco2/impact

Additional Request

All teams participating in the competition are required to perform an additional benchmark for their solution, run on the cluster where the solution was developed. This benchmark should measure both the time to solution and energy consumption over two runs of the algorithm, as it processes the provided additional dataset to identify the cats.

We would love to have a quick chat with you during the competition! Please put together a single slide that highlights the main ideas of your solution, and be ready to give us a 1-minute elevator pitch using it.