Annotation Research in Underwater Cave Environments

Cave Seg Model:

1. Purpose and Importance of Annotations

- **Annotations in Underwater Cave Exploration**: Annotations are used to label specific features in underwater cave images, which helps in training machine learning models for tasks like semantic segmentation. These annotations allow Autonomous Underwater Vehicles (AUVs) and Remotely Operated Vehicles (ROVs) to understand and navigate underwater cave environments more effectively.
- **Significance of Accurate Annotations**: Precise annotations are imporatant as they guide the development of algorithms that detect critical elements like cavelines, obstacles, and open areas. This helps ensure safe and efficient navigation for robotic systems and provides valuable information for cave divers and researchers.

2. Annotation Categories and Object Classes

The project focuses on 13 specific object categories to be annotated in the images, each of which serves a particular purpose in cave navigation and exploration:

- **Caveline**: Represents the path to follow inside the cave and is marked in yellow for maximum contrast. It is essential for navigation and is a key element to identify safe passages and exit routes.
- **First Layer Obstacles**: Immediate areas that should be avoided; these represent the closest potential hazards to the AUV or diver.
- **Second Layer Obstacles**: Areas to be avoided next after the first layer obstacles; these help in planning further maneuvers.
- **Open Area**: Obstacle-free regions where the AUV can navigate safely without risk of collision.
- **Ground Plain**: The cave floor, which helps in spatial orientation and avoiding obstacles on the ground.
- **Scuba Divers**: Marked in magenta, this category is important for cooperative missions where AUVs must navigate alongside human divers, giving them the right of way and avoiding interference.
- **Navigation Aids**: Includes arrows, reels, and cookies that are marked with different colors (e.g., dark red for arrows, red for cookies) and provide directional information or indicate the presence of other divers.
- **Caveline-Attached Rocks**: Indicate attachment points for the caveline and are often associated with changes in direction; marked in brown.

- **Cave Ornaments**: Such as stalactites, stalagmites, and columns, which are also annotated, particularly in caves where these formations are prevalent (e.g., Mexico caves).

3. Annotation Dataset: CaveSeg

- **CaveSeg Dataset**: The dataset used in this project, called CaveSeg, consists of 3,350 pixel-annotated images collected from three cave systems in different geographic locations: the Devil's system in Florida, USA; Dos Ojos Cenote in QR, Mexico; and Cueva del Agua in Murcia, Spain.
- **Annotated Samples**: These images are meticulously labeled to represent the 13 object categories. The annotations involve drawing boundaries around objects and labeling each pixel to correspond to one of the categories mentioned above.
- **Frequency and Distribution**: The dataset shows varied frequency and distribution of these categories. For instance, human divers are present in 40% of the images, while caveline and obstacle-free open areas appear in over 90% of the samples. Navigation markers occupy smaller areas and are present in about 20% of the data.

4. Annotation Challenges

- **Environmental Complexity**: The underwater cave environment presents unique challenges such as poor lighting, visual obstructions, and optical artifacts like blurriness and color distortion due to water properties. These factors complicate the annotation process, as they make it difficult to distinguish between different object categories accurately.
- **Small and Rare Object Categories**: Objects like arrows, cookies, and reels are small and infrequently appear in the dataset. This makes them harder to annotate accurately and requires meticulous attention to detail to ensure that these annotations are useful for training robust models.

5. Annotation Visualization and Tools

- **Sample Visualizations**: Figures in the dataset documentation (e.g., Fig. 2) show examples of the annotated images with corresponding ground truth labels and overlaid visualizations. These visual aids help understand how the annotations look in practice and what each color code represents.
- **Annotation Software**: While the specific annotation tools used are not mentioned, annotations for this type of dataset typically involve using specialized software that supports pixel-level annotation, such as Labelbox, VGG Image Annotator (VIA), or custom-built tools integrated with machine learning libraries like PyTorch.

6. Use of Annotations in Model Training and Validation

- **Training Setups**: Annotations are split into training, validation, and test sets (85:5:10 split ratio) to train and evaluate models like CaveSeg and other state-of-the-art (SOTA) benchmarks. Annotations ensure that the models learn to differentiate between various object categories accurately.
- **Benchmarking and Evaluation**: The quality of annotations directly impacts the model's performance in metrics like mean Intersection Over Union (mIoU), mean class-wise Accuracy (mAcc), and Average pixel Accuracy (aAcc), which are used to assess the model's effectiveness in semantic segmentation tasks.

Annotations in this project are crucial for developing models capable of safely and autonomously navigating underwater caves, providing valuable insights into cave exploration, and ensuring that both robotic and human divers can perform their tasks safely and effectively.

Human-in-loop and Active Learning in Machine Learning:

Human-in-loop Learning:

The human-in-loop process is where human input is used within the machine learning process. This will lead to improvements in the models progress and learning.

Data Labeling and Annotation: Humans can provide accurate labels or annotations for training data, especially when it's complex or ambiguous. This is crucial for supervised learning where labeled data is needed.
Model Training and Tuning: Humans can help by evaluating model performance, identifying areas for improvement, and tuning hyperparameters based on their domain knowledge and expertise.
Error Analysis and Feedback: Human feedback can be used to analyze errors or biases in the model's predictions, helping to refine the model and address issues that automated systems might miss.
Active Learning: This involves a loop where the model identifies examples it's uncertain about, and humans provide labels or corrections. This can make the training process more efficient and focused on areas where the model needs improvement.
Decision Making: In some systems, human judgment is integrated into the decision-making process, where the model's suggestions are reviewed or supplemented by human expertise.

This applies directly to correcting annotations and feeding it back into the model.

Active Learning:

Model comparison, how do differ
How does the same model differ if ran on same data set multiple times

Different Computer Vision Models:

Look at how different computer vision models

Convolutional Neural Networks (CNNs)

- **VGGNet**: Known for its simplicity, VGGNet uses small 3×3 filters throughout the architecture which allows it to go deep (up to 19 layers). It's excellent for feature extraction due to its repetitive stacking of convolutional layers.
- **GoogLeNet (Inception)**: Introduced inception modules that perform multiple convolutions at different scales concurrently, which greatly increases the network's ability to capture information at various scales. It also incorporates dimension-reduction **the** techniques to reduce the computational burden.
- **ResNet**: Revolutionary for its use of residual connections, which allow gradients to flow through the network directly, enabling the training of networks with over a hundred layers by alleviating the vanishing gradient problem.

Region-based Convolutional Neural Networks or in short R-CNNs

- **R-CNN**: Utilizes a selective search to generate region proposals, which are then classified by a CNN. It was a groundbreaking model for showing how deep learning could advance object detection.
- **Fast R-CNN**: Builds on R-CNN by introducing an ROI pooling layer, which significantly speeds up processing by sharing convolutional features across proposed regions.
- **Faster R-CNN**: Adds a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network, enabling almost real-time performance.

- **Mask R-CNN**: Extends Faster R-CNN by adding a branch for predicting segmentation masks on each ROI, making it suitable for tasks requiring instance segmentation.

Yolo (You Only Look Once).

- **YOLOv3**: Balances speed and accuracy effectively, making it suitable for real-time applications. It uses multi-scale predictions and a better class prediction mechanism.
- **YOLOv4**: Improves on YOLOv3 by integrating advanced techniques like mish activation, cross mini-batch normalization, and self-adversarial training to enhance training stability and performance.
- **YOLOv5**: Developed by Ultralytics, it simplifies the architecture and uses PyTorch for more efficient deployment. It continues to improve speed and accuracy for real-time object detection.

Single Shot MultiBox Detector or SSD

- **SSD**: Optimizes for real-time processing by eliminating the need for a separate proposal generation and subsequent pixel or feature resampling stage. It detects objects in a single pass through the detector, using multiple feature maps at different resolutions to capture various object sizes.

U-Net

- **U-Net**: Designed for medical image segmentation, it features an encoder-decoder architecture with a contracting path to capture context and a symmetric expanding path that enables precise localization.

Vision Transformers (ViTs)

- **ViT**: Applies the transformer self-attention mechanism directly to patches of an image, which allows it to consider global context, leading to strong performance in image classification tasks when trained on large datasets.

- **Swin Transformer**: Introduces a hierarchical transformer whose representation is computed with shifted windows, facilitating efficient modeling of various scales and improving performance across multiple vision tasks.

EfficientNet

- **EfficientNet**: Systematically scales the network width, depth, and resolution through a compound coefficient, achieving better efficiency and accuracy compared to other convolutional networks.

Detectron2

- **Detectron2**: A library that implements state-of-the-art object detection algorithms, including Faster R-CNN, Mask R-CNN, and RetinaNet. It is highly modular and customizable, making it a favorite for academic and industrial research projects.

DINO

- **DINO**: Focuses on self-supervised learning by encouraging consistency between different augmentations of the same image, proving effective in learning useful representations without labelled data.

CLIP (Contrastive Language–Image Pretraining)

- **CLIP**: Learns visual concepts from natural language supervision, enabling it to perform a variety of vision tasks using zero-shot capabilities. It leverages a contrastive learning approach between text and images to generalize better across different visual tasks without further training.