**Automated Image Tagging System (AITS)**

**Introduction**

The Automated Image Tagging System (AITS) is an end-to-end solution designed for training and deploying multi-label image classification models. By leveraging AWS SageMaker, S3, and MS COCO datasets, AITS provides a robust pipeline for data preparation, model training, and deployment. The system is designed for scalability and efficiency, enabling high-performance tagging of images into multiple categories.

**Key Features**

1. **Multi-Label Image Classification**:
   * Supports classifying images into multiple categories simultaneously.
   * Utilizes the MS COCO dataset for training, focusing on 5 categories selected from the 80 available.
2. **Customizable Training Pipeline**:
   * Offers parameters such as batch size, learning rate, network depth, and more.
   * Supports transfer learning through pre-trained models.
3. **AWS Integration**:
   * **S3**: For data storage and organization of training, validation, and output files.
   * **SageMaker**: Facilitates distributed model training and hosting for inference.
4. **MLOps Capabilities**:
   * Evaluates trained models with real-world images.
   * Deploys models for scalable and reliable inference.

**System Workflow**

The system's workflow is divided into distinct stages, as outlined below:

**1. Data Collection and Preparation**

* **Dataset**:
  + The MS COCO 2017 validation set is utilized as the primary dataset.
  + The dataset includes 80 categories of objects; 5 are selected for this project.
* **Data Organization**:
  + Images are divided into **training** and **validation** datasets.
  + .lst files are created to map the image data and stored in respective folders:
    - train\_lst: Training image metadata.
    - validation\_lst: Validation image metadata.
* **Storage**:
  + All images and .lst files are uploaded to AWS S3 in the respective folders:
    - train: Training images.
    - validation: Validation images.

**2. Model Training**

The model training phase involves setting up training jobs and configuring hyperparameters.

**Training Job Parameters**

* **Training Instance Count**:
  + Defines the number of instances for training.
  + Supports distributed training when using multiple instances.
* **Training Instance Type**:
  + Typically uses GPU instances (e.g., ml.p3.2xlarge) for optimal performance.
* **Output Path**:
  + Specifies the S3 folder to save training outputs, including the model artifacts.

**Hyperparameters**

* **num\_layers**: Determines the network depth. Example values include:
  + **18 layers**: Lightweight, fast training.
  + **50 layers**: Standard depth for improved accuracy.
  + **152 layers**: Deep networks for complex datasets.
* **use\_pretrained\_model**:
  + Set to 1 to enable transfer learning by leveraging pre-trained weights.
* **image\_shape**:
  + Specifies the dimensions of input images (num\_channels, height, width).
  + Must match the actual dataset image size.
* **num\_classes**:
  + Defines the number of output classes. For this project, it is set to 5.

**Training Process**

* The SageMaker Image Classification Algorithm is used, supporting the application/x-recordio format for handling large datasets.
* The training process adjusts parameters such as learning rate and batch size for optimal results.

**3. Pipeline Integration**

The pipeline is configured to streamline the transition from data preparation to training and deployment:

* Integrates data stored in S3 with the SageMaker training job.
* Configures and executes the training job using the defined parameters and hyperparameters.
* Outputs the trained model to the specified S3 path for further evaluation and deployment.

**4. Model Evaluation and MLOps**

* **Inference**:
  + The trained model is hosted on SageMaker to evaluate its performance with unseen images.
  + The network outputs class probabilities for all target categories.
* **Evaluation Metrics**:
  + Accuracy and loss metrics are used to assess the model's performance.
  + The network achieves significant accuracy, even with only 5 training epochs.
* **Deployment**:
  + The model is deployed for scalable inference, ensuring robustness and reliability.

**Technologies and Tools**

1. **AWS S3**:
   * Provides cloud storage for datasets, training outputs, and .lst files.
   * Ensures high availability and scalability for large datasets.
2. **AWS SageMaker**:
   * Supports distributed training with GPU instances.
   * Enables hosting the model for inference post-training.

**Project Timeline**

The AITS project follows a structured timeline:

| **Week** | **Activity** |
| --- | --- |
| Week 1 | Data Collection and Preparation |
| Week 2 | Initial Model Training |
| Week 3 | Pipeline Integration and Training |
| Week 4 | MLOps and Final Presentation |

**Illustration**

The system employs the following pipeline for effective training and deployment:

1. Data uploaded to S3.
2. SageMaker processes and trains the model using the image classification algorithm.
3. Model artifacts stored in S3 for evaluation.
4. Trained model hosted on SageMaker for inference.

**Conclusion**

The Automated Image Tagging System (AITS) demonstrates a scalable and efficient approach to multi-label image classification. By leveraging AWS SageMaker and S3, the system ensures seamless data processing, model training, and deployment. The integration of MLOps techniques enhances the system's reliability and performance, making it a valuable tool for image tagging applications.

**Team**

* **Faris Hesham**: Developer
* **Kareem Mohamed**: Developer
* **Radwa Sayed**: Developer
* **Noorhan Hamed**: Developer
* **Michel Youssef**: Developer