Performance Work Statement

**Introduction and Background**

**1. Introduction**

Effective decision support systems can help natural resource managers make better decisions in the face of massive amounts of information, multiple competing objectives, and many sources of uncertainty. The USGS Eastern Ecological Research Center, Conte Fish Research Laboratory in Turners Falls, MA has developed a decision support system for streams in the Northeast US, called SHEDS, Spatial Hydro-Ecological Decision Support (ecosheds.org). SHEDS is a web-based system designed to seamlessly link together databases, models, and data visualizations tools with the goal of making environmental data more accessible and easier to use for research, management and decision making. A key component of SHEDS is the Flow Photos Explorer which allows users to predict streamflow from imagery using machine learning models.

**2. Background**

Tools to deliver inputs and outputs of ecological models to natural resource managers are needed for effective management. The SHEDS application was designed to put information in the hands of decision makers, allowing detailed exploration of relationships among ecological drivers at multiple spatial (and eventually temporal) scales.

This scope of work (SOW) is to assist USGS in the development and deployment of a machine learning model to predict streamflow from timelapse images of streams as part of the USGS Flow Photo Explorer (FPE) project. This model, which is being developed by USGS in collaboration with Microsoft AI For Good using the PyTorch machine learning framework, is being trained using an existing database of images and human annotations of pairwise image rankings. The FPE project is currently hosted using Amazon Web Services (AWS) through the USGS Cloud Hosting Solutions (CHS) platform. The infrastructure needed to store and manage the images and data for this project have been developed by USGS and utilize a number of AWS services including Simple Storage Service (S3), Relational Database Service (RDS), API Gateway, Lambda, Batch, and Cognito. The model is being trained and deployed using AWS SageMaker.

The current model development and training process involves a series of steps including data preprocessing, model training, evaluation, and batch inference. Currently, these steps are manually executed as a series of scripts and commands from within a Jupyter notebook using the AWS SageMaker Software Development Kit (SDK) for Python. This process must be repeated for each monitoring station yielding a uniquely trained model for each station (FPE currently has over 200 stations). To improve the efficiency of this process and increase the scalability of the FPE project for training this model across many stations, USGS is seeking assistance to automate the model training and deployment process, and to establish a process for deploying this model to support real-time inference.

To assist in the development and deployment of this model, this SOW consists of two tasks: 1) assist in the development of an automated machine learning pipeline that will facilitate model training and inference on a station-by-station basis, and 2) to develop a real-time inference endpoint of a trained model specific to a single station that can be used to predict streamflow from new images transmitted to FPE through cellular or satellite connections.

For both tasks, the contractor will focus solely on developing the necessary components within AWS SageMaker, namely a SageMaker Model Building Pipeline and a SageMaker Real-time Inference Endpoint (or possibly using alternative AWS computing service such as EC2 or Batch). The USGS will be responsible for developing the remaining project infrastructure to handle data ingest, storage, and access control, much of which already exists.

**3. Place and Period of Performance**

a. Work can be performed on-site or at the USGS Eastern Ecological Science Center, Conte Research Laboratory, or through remote telework from home through a mutually agreeable telework agreement.

b. Period of performance is one year from date of award.

**4. Work Requirements**

1. Technical Requirements.
   1. **Task 1: Automated Machine Learning Pipeline**

The first task of this SOW is to assist in the development of an automated machine learning pipeline that will facilitate model training and batch inference. Currently, model training involves a manual process consisting of input data pre-processing, model training, performance evaluation, and lastly prediction (i.e., batch inference). The goal of this task is to develop an automated pipeline for performing these sequential steps based on a single set of input configurations such as the station ID, model hyperparameters, and other training or inference variables. This pipeline will be designed to efficiently train and apply the FPE model on many individual sites. This pipeline will be implemented using AWS SageMaker Model Building Pipelines to integrate with the existing FPE infrastructure. The contractor will not be responsible for developing the associated infrastructure to store, manage, and access the data and images needed for training. The contractor will only be responsible for developing the Model Building Pipeline based on the existing scripts and code that are currently being executed manually.

* 1. **Task 2: Real-Time Inference Endpoint**

The second task of this SOW is to develop a real-time inference endpoint that can be used to predict streamflow from new images transmitted to FPE through cellular or satellite connections. Currently, predictions from the FPE model are generated using batch inference on the full set of existing images at each site. USGS is currently working with various collaborators to develop methods for ingesting images in real-time. The goal of this task is to develop a model endpoint that can be used to generate predictions on these images as they are received, and to save the results to the FPE database. This endpoint will be implemented as a Real-time Inference Endpoint in AWS SageMaker or, if necessary, another AWS computing service such as AWS EC2 or Batch. The contractor will be solely responsible for developing the process and associated code necessary to deploy a trained model (using the pipeline created in Task 1) as a real-time inference endpoint. The contractor will not be responsible for developing the infrastructure to handle real-time data and image ingest that would trigger the invocation of the real-time endpoint for generating a model prediction of each image.

1. **Criteria for acceptance of Product or Service**. Compliance for this service will be based on 1) successful completion of tasks outlined above including fully functioning models and pipelines developed in consultation with USGS personnel and integrated into existing code and pipelines. The code will be freely available from a code repository such as GitHub.
2. **Deliverables**
   1. Tasks 1-2. Fully functioning code and pipelines as described for each task 1-2.
      1. Task 1
         1. AWS SageMaker Model Building Pipeline and associated Python code for FPE model training and prediction. The pipeline will be defined using the SageMaker Pipelines SDK and/or pipeline definition JSON schema. The Python code executed at each step in the pipeline will be developed by adapting existing Python code for data preprocessing, training, evaluation, and batch inference.
         2. Pipeline inputs include a station ID and a dictionary of configuration variables and hyperparameters for controlling data pre-processing and model training. The Pipeline will use the station ID to retrieve metadata and a list of available images from an existing database (RDS instance). The image dataset will include the location of the image files stored in an existing S3 bucket.
         3. The Pipeline will persist each training run – saving model inputs and hyperparameters, model artifacts and learned parameters, model evaluation metrics, and inference predictions to existing FPE infrastructure (AWS RDS instance and S3 storage bucket).
         4. The Pipeline will be designed to be invoked for handling only one model (station) at a time, and thus will not be required to scale to multiple simultaneous model trainings.
         5. This deliverable will be integrated by USGS into the existing FPE infrastructure, and thus not require the development of any other AWS resources.
      2. Task 2
         1. A set of Python scripts or templates for setting up one or more AWS services to support real-time inference at up to 10 stations using trained models (one model per station) from Task 1.
         2. For each station/model, the endpoint is expected to be invoked no more than once every 15 minutes.
         3. The AWS services for this task may include one or more AWS SageMaker Real-time Inference Endpoints and/or other AWS services such as AWS Batch or EC2 that could be designed to provide real-time inference across multiple stations and models (up to 10).
         4. The deliverable will include the ability to configure the AWS service(s) to provide real-time inference based on a set of trained models (using artifacts stored in S3 and as generated by the pipeline in Task 1), which will accept a timelapse photo as the sole input.
         5. The endpoint will return a response containing the predicted value for the provided photo.
         6. The endpoint will not require authentication/access for external users as it will only be invoked by internal resources (e.g., an existing Lambda function).
         7. This task will focus solely on the AWS service(s) needed to perform real-time inference using the trained models for multiple stations. This task will not involve developing the infrastructure or code needed to handle image ingest or output storage. The service or endpoint will be invoked by existing FPE infrastructure (e.g., lambda function) managed by USGS.
3. **Desired Experience and Capabilities**
   1. Statistical analysis & modeling, data science
   2. Machine learning & artificial intelligence
   3. Scientific data analysis
   4. Familiarity with AWS SageMaker, Batch, and EC2
   5. Knowledge of Python programming languages.