Vegetation Specific Fire Assessment

In this exercise, we would like to go through an existing system and improve it with new functionality. If something is not specified feel free to make assumptions and add them as part of the context.

Problem & Context

Our current Forest Fire Alarm Monitoring System (FFAMS - 🌲🔥🚨✉) is currently able to detect fires accurately if the aerial images (min. 100m x 100m) we receive via our Django-powered HTTP API come from a known location where we expect specific types of vegetation to exist.

We are therefore restricted to send our email alerts only to fire brigades within those areas and we cannot scale our system to other geographic regions/forests.

Our research team has developed both a:

- Vegetation Classifier 🍁🍃🌴 - which categorizes which **types** of vegetation can be seen in the image and
- Vegetation type-specific fire risk scoring models.

We need to integrate these new elements into our existing system in order to be able to geographically expand our operation.

The Vegetation Classifier is currently deployed as an independent HTTP API. Given an image, it synchronously returns a set of vegetation types and corresponding weights found in the image. The vegetation types are identified by a unique string identifier e.g. DRY_EVERGREEN_FOREST, TROPICAL_FOREST and the API returns a JSON with a mapping:

```
{
    DRY_EVERGREEN_FOREST: 0.2344,
    TROPICAL_FOREST: 0.1112,
    ...
}
```

The new scoring models are Tensorflow .pb models that require CPU intensive operations. Each model takes approximately the same time as our current mechanism (pure Python colour analysis algorithm) to calculate the score. We don't want our overall processing time to increase.

We would like to output a new overall fire-likelihood score based on a **weighted average** of the results of the image for each of these vegetation-specific models. The weights in this weighted average are determined by the classifier results described above. Finally, we trigger an alert depending on whether this new global score is over a specific risk fire threshold.

We also need to reassess our scaling strategy since currently, we have a fixed setup with a pair of servers for our single monolithic FAMMS application and the new Vegetation Classifier.

Your Task

You have 2 deliverables:

1. **A markdown document with the technical design of the implementation.** Fill in the blank sections in the rest of the document below, answering the questions as appropriate.
2. **An implementation for the code for running the vegetation-specific models.** Essentially you must define a function:

```
is_it_about_to_catch_fire(image: Image, classification_scores:
Dict[str, float]) -> bool
```

Here `image` represents the image pixels and `classification_scores` is a dictionary with the classification probability for each of the vegetation classes. You should output a yes/no decision on whether to call the fire brigade or not. 🚒 The concrete implementation will depend on your design choices. You should add unit tests for your implementation. For these, you should fake/mock any external system calls you might have introduced to ensure the tests are runnable locally. **The implementation may be in any language of your choice.**

Please submit your solutions as a link to a git repo on github, bitbucket, gitlab or other code hosting platform.

You should aim to spend 75% of your time on the write up and 25% on the code implementation. We are assessing your ability to come up with a solid technical design, and you should see the code as complementing the document, rather than the other way round. However, the code should work and be runnable and testable locally.

DESIGN TEMPLATE FOLLOWS

Solution

How are you intending to solve this problem? Start with an overview/broad strokes.

Architecture

- *What does this service or feature look like? Please draw diagrams where possible*
- *Why does it look like that?*
- *Consider the service/feature boundaries; what business capability is this service responsible for?*
- *What data does it own?*

<YOUR ANSWER>

Dependencies & Integration

How does it interact with existing services/functionality/components, both upstream and downstream? Consider both existing and planned software and services.

<YOUR ANSWER>

Interfaces

What interfaces and operations might this expose?

<YOUR ANSWER>

Infrastructure

What pieces of infrastructure are you using/reusing? E.g. Amazon service, open source libraries, databases, programming languages? Which dependencies are you introducing?

<YOUR ANSWER>

Scale & Performance

How does the feature scale?

- *What are the expected scale characteristics (at least back of an envelope) for this service? Consider CPU, memory and storage. Can it horizontally scale?*
- *How might your design decisions increase/reduce cost?*

<YOUR ANSWER>

Reliability

What level of reliability are you aiming for? If client-side software, what range of devices and OS versions are you aiming to test on and support? If a service, are you comfortable achieving 99.95% (or higher) availability? Is the software experimental or aiming to support thousands-millions of users straightaway?

<YOUR ANSWER>

Redundancy

How do you manage backups and restores, if needed? Can you handle the loss of connectivity? Are there alternative services and fallbacks? What are your hard RAM/disk limits, if any?

<YOUR ANSWER>

Monitoring & Instrumentation

How will you understand the behaviour of this feature/service in production - especially how it meets product goals / provides customer value? Will there be any alerting or metrics that you will measure? How will you monitor the technical behaviour of your service/feature?

<YOUR ANSWER>

Failure Scenarios

When will this design/feature/approach fail? How will you mitigate the impact of these failures?

<YOUR ANSWER>

Risks & Open Questions

What are the major risks that might prevent your software from working or be successful? What don't you know yet that might change this design or how you approach implementation?

<YOUR ANSWER>