1. Environment Setup: Describe how you would set up a machine learning environment from scratch without any pre-existing tools or infrastructure.

   Throughout the project there will be different stages, thus, different environments and tools are needed. For experimentation and development, I would use jupyter notebooks and prefer VSCode as the IDE. I would use GitHub for development lifecycle and set up a CI/CD pipeline that automatically checks the suitability or cleanness of the code. For the training, I would use Azure Machine Learning as the cloud platform and rent compute instances over there. I would use the same platform throughout the project for other phases like deployment and data storage. When it comes to data storages, it is essential to use SSDs because the model will query the data frequently. Thus, the datastore should have low latency and the connection between compute instance should be fast. I would use Azure Cosmos DB which is a NoSQL database. On top of the normal database there needs to be a vector database that enables us to do efficient vector search (similarity search). For this we can use Azure AI Search.

2. Data Acquisition: Explain your strategy for scraping the internet and other relevant sources to gather data on celebrities wearing Vlisco designs. How would you ensure the data includes both past and current celebrities?

   I would start by doing keyword searching over Google Images and Pinterest. I would search for keywords like "African fashion Vlisco", "Festival Vlisco", "Gala Vlisco", "African Clothing". A result from a single search would be associated to other keywords so I would use those keywords to search over as well. I would save images from each search, together with its annotations. The relationship between the annotation and image can be used as a direct indication Vlisco design or celebrity. Then I would use YOLOv8, an open-source object detection model, to locate and crop humans in images (discard the image if there isn't any humans).

   When it comes to face recognition, I would use the face_recognition library to detect and crop faces in an image and create embeddings of faces. I would apply locate-crop-embed sequence to embed faces in ms-celeb (that has 1 million images of celebrities) and CASIA-Face-Africa datasets, both designed for face recognition tasks. By comparing the embeddings of ms-celeb to CASIA-Face-Africa I would create a dataset of faces of African celebrities.

   After scraping Google Images and Pinterest I would use the same approach (keyword search) on Instagram and Twitter and after those I would move to scraping the archives of fashion magazines and blogs.

   Obtaining current data is easier than the past I say, because there are more sources for today's events, and they are easier to access. After doing some research I've found out that there are online sources offering fashion-related images dating back to 1890s. These are direct archives of fashion magazines like The Vouge Archive which has digitized versions of their publications or organizations that aim to keep vintage fashion alive like The Vintage Fashion Guild.

   I would apply the same pipeline to those images. Locate and crop humans, locate faces, create embeddings of faces and then find celebrities based on similarity. For identifying celebrities from past I would create a dataset of historical figures collected from Historical Figures section of Google Arts & Culture which offers various images of important figures grouped by years.

3. Preprocessing and Labeling: Outline the preprocessing steps you would take to clean and prepare the scraped data for analysis. How would you label the data to indicate which images feature celebrities wearing authentic Vlisco designs?

As mentioned above, First I would use YOLOv8 do detect and crop humans in the images. Then, I would convert every image into PNG format. After that I will resize the images to make them the same size. Then, I will normalize the pixel values to -1 and 1. Lastly, I would create a data augmentation pipeline where images are slightly rotated and mirrored.

For labeling there can be two approaches. For both of them, I would use the images that are in the catalogues of Vlisco. The catalogue would provide photos of designs from different angels.

1) I would use a pre-trained model like VGG16 and remove its classification layers so that the model's output would be the embeddings (features) of inputs. The choice of VGG16 comes from the fact that it is a pre-trained model on a diverse set of classes. Thus, its earlier layers will be able to extract geometrical patterns. This aligns with the designs of Vlisco. When I looked at the designs, I saw some sort of geometrical patterns like spirals squares etc. Thus, the model will be able to create accurate embeddings of the designs. For a better accuracy I would fine-tune the model on the dataset formed from the catalogue. Then I would feed the scraped dataset to the model to classify them as Vlisco designs or not. Here, we can take an iterative approach to make the model even better by iteratively expanding the catalogue dataset by adding the most %5 confident guesses and re-training the model on this extended dataset.
2) Another approach could be taking the catalogue set and fine-tuning stable diffusion 2.1 model such that the model will be able to create synthetic catalogue images of Vlisco designs. Then I would train a model on this dataset to create the embeddings.

In both cases the classification is based on the similarity between embeddings of authentic Vlisco designs (coming from catalouge) and scraped data.

4. Model Development: Propose a machine learning approach for detecting Vlisco-wearing celebrities in the dataset. This could involve techniques such as image classification, object detection, or face recognition. Explain your choice of algorithm and how you would train and evaluate the model.

   As I already mentioned, I would fine-tune VGG16 on catalogue data and then use it to create embeddings of both scraped and catalogue data and then decide based on embedding similarity (cosine similarity). The idea is to make use of the geometric patterns that the Vlisco designs have. The approaches may differ; however, the main idea is to teach the model about the characteristics of the available designs. Use the model as a feature extractor and then compare the similarities of extracted features between valid designs and collected data. Face recognition will be implemented as discussed above, after concluding if an image has a Vlisco design or not.

5. Deployment and Maintenance: Consider how you would deploy the trained model in a production environment and ensure its ongoing performance and reliability throughout the internship.

   I would first convert the trained model into a format that is compatible with different hardware and drivers. For this I would use ONNX Runtime library. Then, I would use Docker to containerize the converted model along with its pre-processing pipeline, dependencies, and API handler. Then this container image will be deployed to the compute instances rented on Azure Machine Learning. However, the details of deployment depends on the use case.

1) Online deployment where the system constantly listens to Instagram, googled trends, twitter etc. to scrape data. Whenever it gets triggered from a keyword or hashtag or by processing a newly scraped image, it triggers an alert: Vlisco is detected there. Here the scraped data can be accumulated on the database and then queued for model. Here the processes should be held in parallel with multiple compute instances to ensure real-time performance.

2) Offline deployment where model has to process a dataset that is already available. The model starts from the earliest dated photo and starts processing it.

For both scenarios the scalability and distribution of the work is crucial. To account for it, I would use Kubernetes to manage the deployment of containers based on GPU usage and the amount of queued data. On top of it, I would create a dashboard for statistics that shows the triggers, alarms, downtimes, GPU utilization etc to monitor the deployment and performance.

Mustafa Talha Ilerisoy