Air Quality Index (AQI) Prediction System

1. Core Concept

This system uses deep learning to estimate Air Quality Index (AQI) from satellite imagery by analyzing haze features and visual characteristics.

2. Technical Components

2.1 AQI Estimation Model

```
class AQIEstimator(nn.Module):
    # CNN architecture:
    - Input layer: 3 channels (RGB)
    - Conv layers: 32 -> 64 -> 128 filters
    - Dense layers: [feature_size + 6] -> 256 -> 64 -> 1
    - Uses ReLU activation and dropout
```

2.2 Haze Feature Extraction

The system extracts six key features:

1. Dark Channel Mean
2. Transmission Mean
3. Atmospheric Light
4. Contrast
5. Saturation
6. Haze Density

3. Usage Example

```
# Initialize predictor
predictor = AQIPredictor()

# Extract haze features
extractor = HazeFeatureExtractor()
features = extractor.extract_features(image)

# Predict AQI
```

```
aqi = predictor.predict_aqi(image, features)
print(f"Estimated AQI: {aqi}")
```

4. Input Processing

- Image preprocessing:
  - Resized to 256x256
  - Normalized to [0,1]
  - Converted to tensor format
- Feature processing:

  ```
  feature_values = [
      dark_channel_mean,
      transmission_mean,
      atmospheric_light,
      contrast,
      saturation,
      haze_density
  ]
  ```

5. Technical Requirements

- Python 3.7+
- PyTorch
- OpenCV
- NumPy
- CUDA (optional)

6. Model Architecture

```
Input Image (3x256x256)
│
├─► Conv1 (32 filters) ─► MaxPool ─► ReLU
│
├─► Conv2 (64 filters) ─► MaxPool ─► ReLU
│
├─► Conv3 (128 filters) ─► MaxPool ─► ReLU
│
├─► Flatten
│   │
```

```
│    └─► Concatenate with haze features
│
├─► Dense (256) ─► ReLU ─► Dropout
│
├─► Dense (64) ─► ReLU ─► Dropout
│
└─► Dense (1) ─► AQI Prediction
```