

Advanced Stable Diffusion WebUI with Face Processing

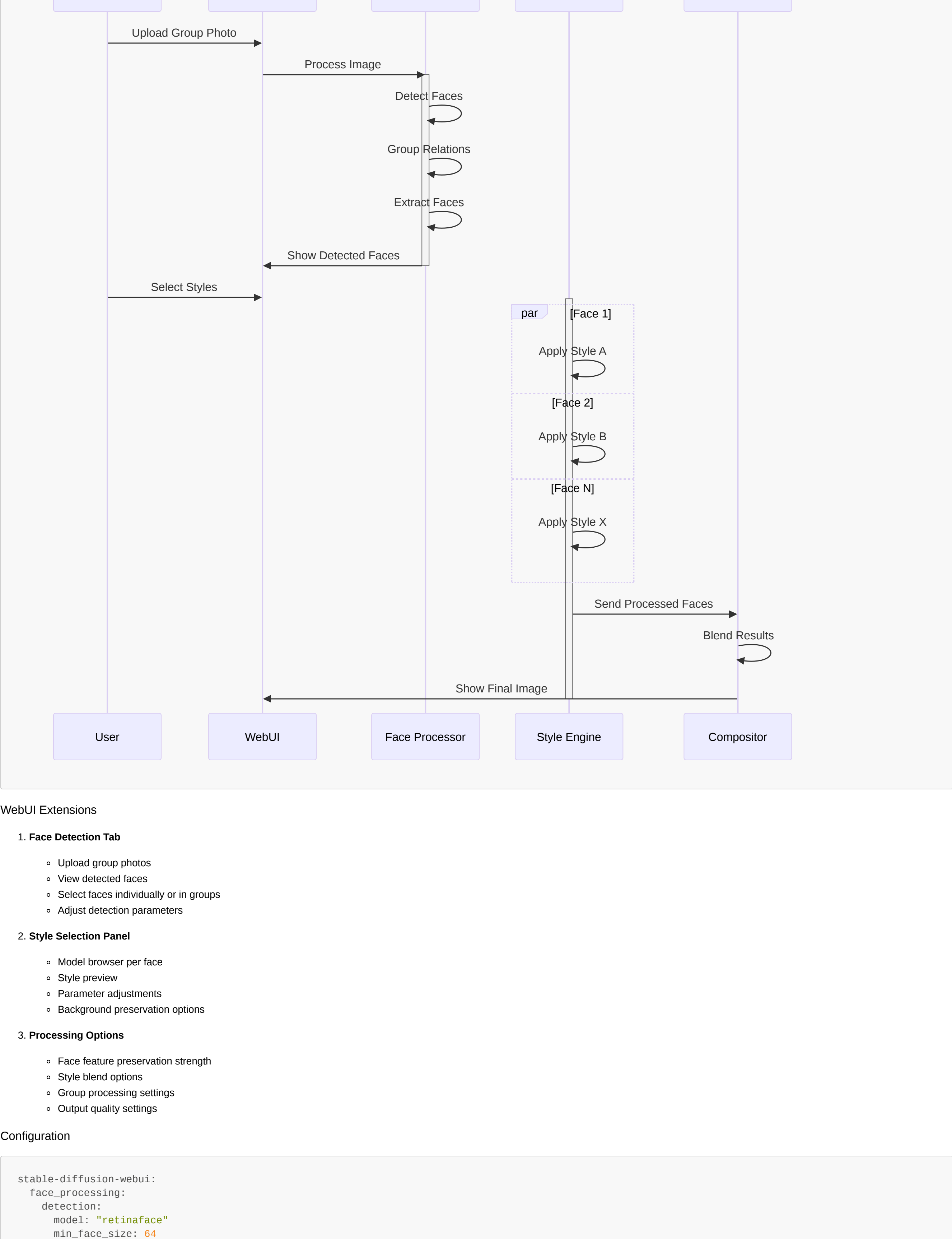
System Overview

An extension of stable-diffusion-webui that adds advanced face processing capabilities, allowing users to:

- Detect and process individual faces in group photos
- Apply different style models to each face
- Preserve facial features and relationships
- Handle family/group photos intelligently

Core Components

Face Processing Pipeline



WebUI Extensions

1. Face Detection Tab

- Upload group photos
- View detected faces
- Select faces individually or in groups
- Adjust detection parameters

2. Style Selection Panel

- Model browser per face
- Style preview
- Parameter adjustments
- Background preservation options

3. Processing Options

- Face feature preservation strength
- Style blend options
- Group processing settings
- Output quality settings

Configuration

```
stable-diffusion-webui:
  face_processing:
    detection:
      model: "retinaface"
      min_face_size: 64
      confidence_threshold: 0.95
    recognition:
      model: "arcface"
      clustering: true
      relationship_threshold: 0.6
    extraction:
      preserve_features: true
      background_handling: "smart"
  style_transfer:
    models_path: "/models"
    cache_size: 4
    parallel_processing: true
    blend_method: "poisson"
```

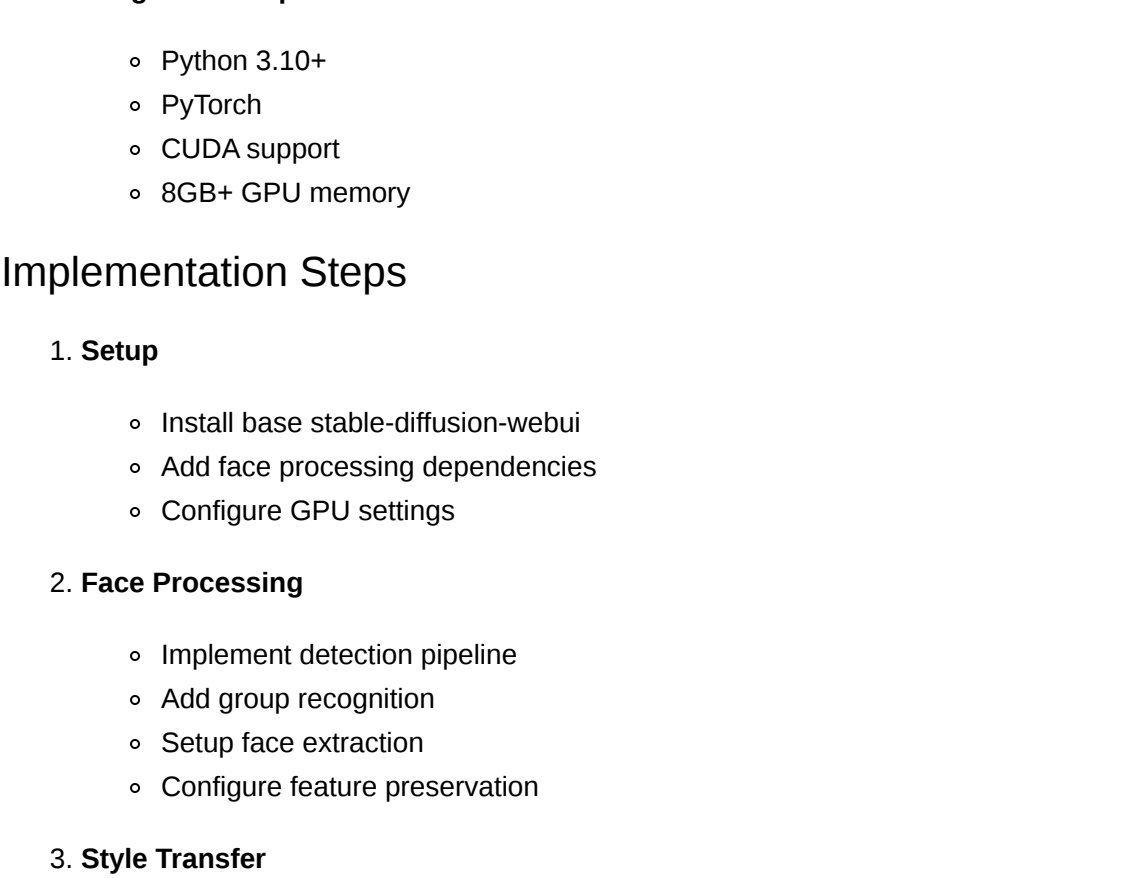
Extended API

```
# Face Processing
POST /sdapi/v1/face/detect
POST /sdapi/v1/face/extract
POST /sdapi/v1/face/group

# Style Application
POST /sdapi/v1/style/preview
POST /sdapi/v1/style/apply
POST /sdapi/v1/style/blend
```

User Interface

Main Interface



Technical Components

1. Face Processing

- RetinaFace for detection
- ArcFace for recognition
- BiSeNet for segmentation
- DLIB for landmarks

2. Style Application

- Per-face style transfer
- Background preservation
- Feature preservation
- Result blending

3. Integration Requirements

- Python 3.10+
- PyTorch
- CUDA support
- 8GB+ GPU memory

Implementation Steps

1. Setup

- Install base stable-diffusion-webui
- Add face processing dependencies
- Configure GPU settings

2. Face Processing

- Implement detection pipeline
- Add group recognition
- Setup face extraction
- Configure feature preservation

3. Style Transfer

- Integrate style models
- Add per-face processing
- Implement blending
- Optimize performance

4. UI Integration

- Add face detection tab
- Create style selection interface
- Implement preview system
- Add processing controls

Ubuntu Implementation

System Requirements

```
# Minimum Hardware Requirements
CPU: 4 cores (Intel i5/Ryzen 5 or better)
RAM: 16GB
GPU: NVIDIA with 8GB VRAM (RTX 3060 or better)
Storage: 50GB SSD

# Recommended Hardware
CPU: 8 cores (Intel i7/Ryzen 7 or better)
RAM: 32GB
GPU: NVIDIA with 12GB VRAM (RTX 3080 or better)
Storage: 100GB NVMe SSD
```

Installation Steps

```
# 1. Update system
sudo apt update && sudo apt upgrade -y

# 2. Install NVIDIA drivers
sudo add-apt-repository ppa:graphics-drivers/ppa
sudo apt update
sudo apt install nvidia-driver-535 nvidia-cuda-toolkit

# 3. Install Python and dependencies
sudo apt install python3.10 python3.10-venv python3-pip git

# 4. Install system libraries
sudo apt install libgl1-mesa-glx libgl1-mesa-dev wget

# 5. Clone and setup stable-diffusion-webui
git clone https://github.com/AUTOMATIC111/stable-diffusion-webui
cd stable-diffusion-webui
source venv/bin/activate

# 6. Install face processing requirements
pip install insightface==0.7.3
pip install onnxruntime-gpu==1.15.1
pip install dlib==19.24.2
pip install opencv-python==4.8.1.78
```

Launch Script

```
#!/bin/bash
# Launch.sh
export CUDA_VISIBLE_DEVICES=0
source venv/bin/activate
python launch.py --api --listen --enable-insecure-extension-access
```

Real Hardware Costs (Approximate in MAD)

Minimum Setup

| Component | Model | Cost (MAD) |
|-------------|---------------|------------|
| CPU | Ryzen 5 5600X | 2,100 |
| Motherboard | B550M | 1,000 |
| RAM | 16GB DDR4 | 500 |
| GPU | RTX 3060 12GB | 3,500 |
| SSD | 500GB NVMe | 500 |
| PSU | 650W Gold | 700 |
| Case | Basic ATX | 500 |
| Total | | 6,800 MAD |

Recommended Setup

| Component | Model | Cost (MAD) |
|-------------|---------------|------------|
| CPU | Ryzen 7 5800X | 3,200 |
| Motherboard | X570 | 1,800 |
| RAM | 32GB DDR4 | 900 |
| GPU | RTX 3080 12GB | 7,000 |
| SSD | 1TB NVMe | 900 |
| PSU | 850W Platinum | 2,200 |
| Case | Good Airflow | 900 |
| Total | | 15,900 MAD |

Power Consumption & Running Costs

Daily Usage (8 hours/day)

| Component | Minimum Setup | Recommended Setup |
|------------------|---------------|-------------------|
| System Idle | 10W | 15W |
| Under Load | 45W | 65W |
| Average Usage | 27W | 40W |
| Daily Power (8h) | 2.2 kWh | 3.2 kWh |
| Monthly Power | 66 kWh | 96 kWh |
| Monthly Cost* | -88 MAD | -115 MAD |

* Based on Moroccan electricity rates (~1.2 MAD/kWh)

Performance Metrics

Processing Times (Approximate)

| Operation | Min Setup | Recommended |
|------------------|-----------|-------------|
| Face Detection | 8-15s | 0-5s |
| Style Preview | 2-5s | 1-5s |
| Full Processing | 8-15s | 5-7s |
| Batch Processing | 15s/image | 9s/image |

Maintenance Considerations

1. Cooling Requirements

- Good airflow case
- Room temperature < 26°C
- Regular dust cleaning
- GPU thermal paste renewal yearly

2. Software Updates

- Regular system updates
- CUDA toolkit updates
- Model updates
- Dependencies updates

3. Backup Strategy

- Model backups
- Configuration backups
- Generated images backup
- System state backup

Server Implementation

Server Requirements Analysis

Can this run as a server? Yes, with the following considerations:

- The system uses REST APIs already
- Processing can be done asynchronously
- Multiple requests can be queued and processed
- Resource scaling is straightforward

Ubuntu Server Setup

Hardware Requirements (Production Server)

| Component | Specifications | Monthly Cost (MAD) |
|------------------------|---------------------------|--------------------|
| CPU | AMD EPYC 7443P (24 Cores) | Part of server |
| RAM | 64GB ECC | Part of server |
| GPU | NVIDIA A4000 (16GB) | Part of server |
| Storage | 2TB NVMe SSD | Part of server |
| Dedicated Server (OVH) | All included | ~4,500 MAD |
| Power & Cooling | Included | Included |
| Internet (10Gbps) | Included | Included |

Server Installation

```
# 1. Update Ubuntu Server
sudo apt update && sudo apt upgrade -y

# 2. Install required packages
sudo apt install -y \
  python3.10 \
  python3.10-venv \
  python3-pip \
  git \
  nginx \
  supervisor \
  libgl1-mesa-glx \
  libgl1-mesa-dev

# 3. Install NVIDIA components
sudo apt install -y nvidia-driver-535 nvidia-cuda-toolkit

# 4. Setup application
git clone https://github.com/AUTOMATIC111/stable-diffusion-webui
cd stable-diffusion-webui
python3 -m venv venv
source venv/bin/activate

# 5. Install dependencies
pip install -r requirements.txt
pip install insightface==0.7.3 onnxruntime-gpu==1.15.1 dlib==19.24.2

# 6. Setup Supervisor
sudo nano /etc/supervisor/conf.d/stable-diffusion.conf
```

Supervisor Configuration

```
[program:stable-diffusion]
command=/home/ubuntu/stable-diffusion-webui/venv/bin/python launch.py --api --listen --enable-insecure-extension-access
directory=/home/ubuntu/stable-diffusion-webui
user=ubuntu
autorestart=true
stderr_logfiles=/var/log/stable-diffusion.err.log
stdout_logfiles=/var/log/stable-diffusion.out.log
environment=CUDA_VISIBLE_DEVICES="0"
```

Nginx Configuration

```
server {
    listen 80;
    server_name your_domain.com;

    location / {
        proxy_pass http://127.0.0.1:7860;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        client_max_body_size 20M;
        proxy_read_timeout 300s;
    }
}
```

Real Server Costs (Monthly in MAD)

Dedicated Server Option

| Provider | Specs | Cost (MAD/month) |
|----------|--|------------------|
| OVH | AMD EPYC 7443P 64GB RAM NVIDIA A4000 2TB NVMe 10Gbps unmetered | ~4,500 |
| Hetzner | Similar specs | ~4,800 |
| Scaleway | Similar specs | ~5,200 |

Performance Capacity

| Metric | Value |
|-----------------------------|-----------------|
| Concurrent Users | 20-30 |
| Requests per Hour | 200-250 |
| Average Process Time | 5-8 seconds |
| Max Image Size | 4096x4096 |
| Monthly Processing Capacity | ~100,000 images |

Additional Costs

| Item | Cost (MAD/month) |
|------------------------|---------------------------|
| Domain Name | ~100 |
| SSL Certificate | Free (Let's Encrypt) |
| Backup Storage (1TB) | ~200 |
| DDoS Protection | Included |
| Monitoring Tools | Free (Prometheus/Grafana) |
| Total Additional Costs | ~300 |

Server Maintenance

1. Daily Tasks

```
# Check system health
sudo supervisorctl status
df -h

# Monitor logs
tail -f /var/log/stable-diffusion.err.log
```

2. Weekly Tasks

```
# Update system
sudo apt update && sudo apt upgrade -y

# Cleanup old images
find /path/to/images -mtime +7 -delete

# Backup configuration
tar -czf config-backup.tar.gz /path/to/config
```

3. Monthly Tasks

```
# Full system backup
# Model updates
# Performance optimization
# Log rotation
```

Scaling Considerations

1. Vertical Scaling

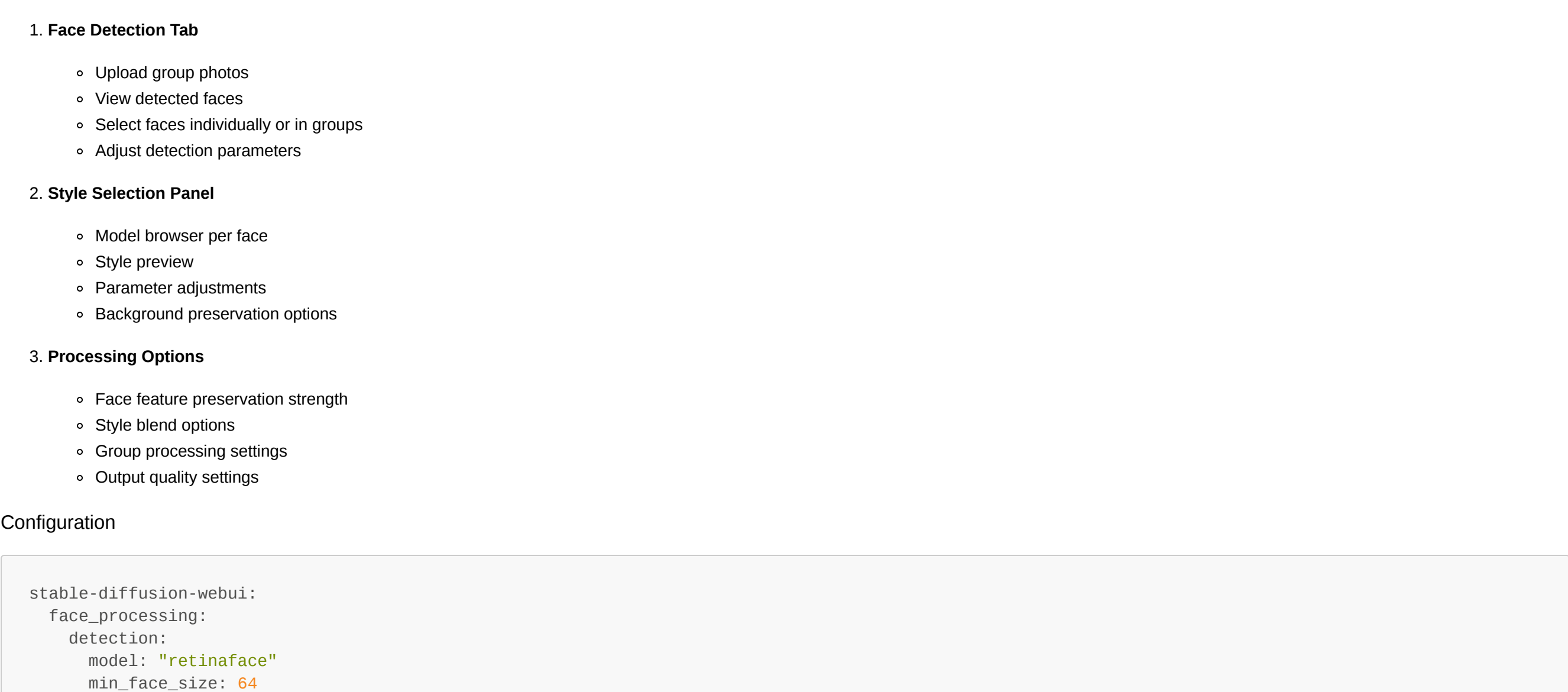
- Upgrade to A5000 GPU (~2,000 MAD/month)
- Increase RAM to 128GB (~500 MAD/month)
- Faster CPU options available

2. Horizontal Scaling

- Add additional servers
- Load balancer setup (~300 MAD/month)
- Distributed storage needed

Server Resource Monitoring

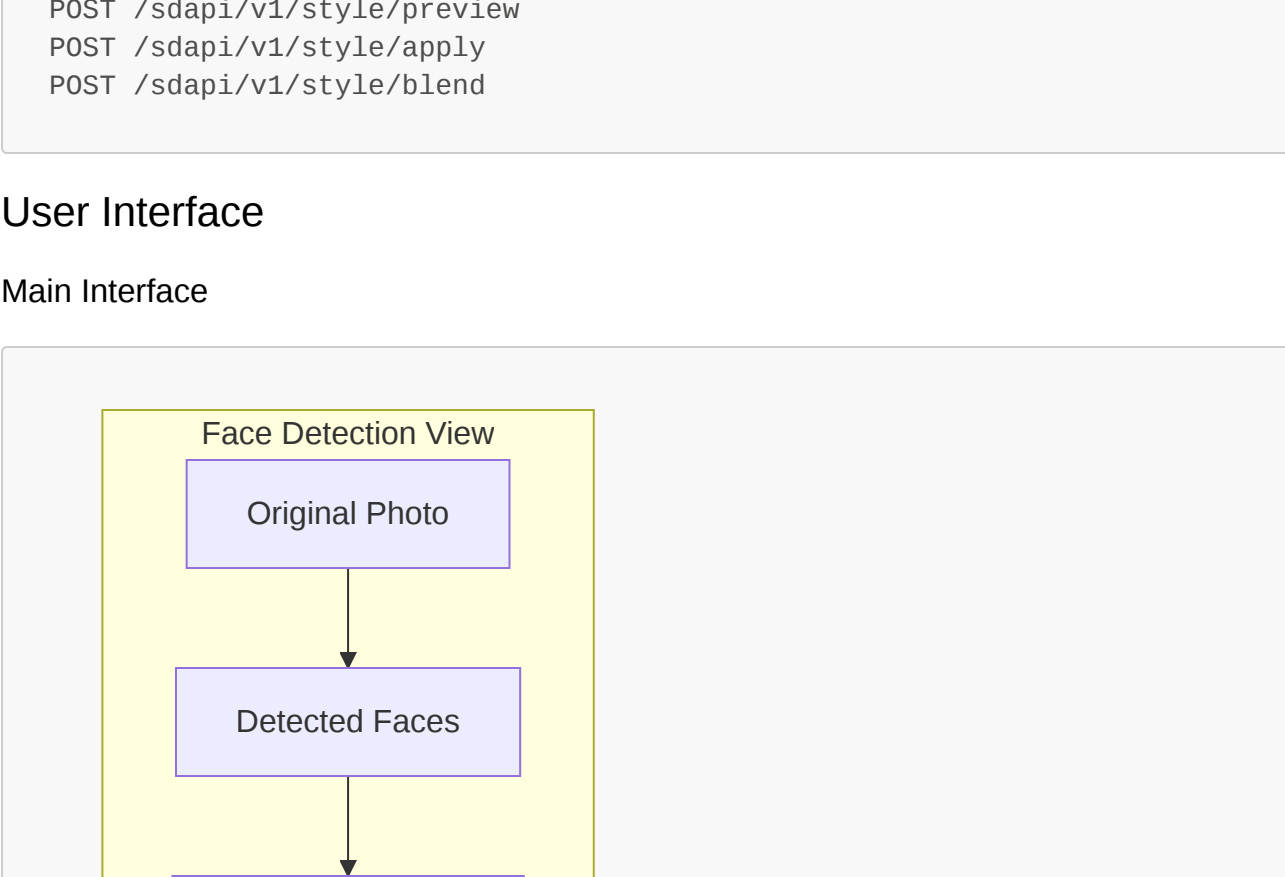
Request Processing Flow



System Health Dashboard



Cost Comparison



Server Monitoring Configuration

```
# Prometheus configuration with high contrast
global:
  scrape_interval: 15s
  evaluation_interval: 15s
  external_labels:
    monitor: 'stable-diffusion-monitor'

scrape_configs:
  - job_name: 'stable-diffusion'
    static_configs:
      - targets: ['localhost:7860']

  - job_name: 'node-exporter'
    static_configs:
      - targets: ['localhost:9100']

  - job_name: 'gpu-metrics'
    static_configs:
      - targets: ['localhost:9835']
```

Performance Dashboard Layout

