NVIDIA Jetson Nano Configuration Guide

PD Auto-Processing System Setup

🔧 Hardware Specifications

- **Board**: NVIDIA Jetson Nano Developer Kit (4GB)
- **RAM**: 4GB LPDDR4
- **Storage**: 32GB microSD card
- **GPIO**: 40-pin header for LED control
- **Network**: WiFi/Ethernet capability
- **Power**: 5V 4A power adapter (barrel jack)

📋 Pre-Setup Requirements

Required Items:

- ✅ NVIDIA Jetson Nano Developer Kit
- ✅ 32GB microSD card (Class 10, UHS-I recommended)
- ✅ 5V 4A power adapter with barrel jack
- ✅ HDMI monitor + cable
- ✅ USB keyboard and mouse
- ✅ Ethernet cable (for initial setup)
- ✅ Computer with SD card reader

🚀 Step 1: SD Card Preparation

1.1 Download JetPack

```bash
# Download from NVIDIA Developer website
# JetPack 4.6.1 (recommended for stability)
# File: jetson-nano-jp461-sd-card-image.zip
# Size: ~6.5GB
```

1.2 Flash SD Card

```bash
```

```
# Using Balena Etcher (recommended)
1. Download Balena Etcher
2. Insert 32GB SD card
3. Select downloaded .img file
4. Flash to SD card (takes ~15-20 minutes)
5. Safely eject SD card
```

1.3 SD Card Partitioning (Optional for 32GB)

```bash
# After first boot, expand filesystem if needed
sudo resize2fs /dev/mmcblk0p1
```

🔌 Step 2: Initial Hardware Setup

2.1 Physical Connections

```bash
# Connect in this order:
1. Insert flashed 32GB SD card
2. Connect HDMI monitor
3. Connect USB keyboard and mouse
4. Connect Ethernet cable
5. Connect 5V 4A power adapter (LAST)
```

2.2 First Boot Configuration

```bash
# Follow on-screen setup wizard:
1. Select language and keyboard layout
2. Create user account:
   Username: jetson (recommended)
   Password: [your-secure-password]
3. Set timezone
4. Connect to WiFi (if available)
5. Complete setup (5-10 minutes)
```

🌐 Step 3: Network Configuration

3.1 WiFi Setup

bash

```bash
# Configure WiFi for your network
sudo nmcli device wifi connect "Darahas's Vivo V23" password "naga@4321"

# Verify connection
ip addr show wlan0
ping google.com

# Set static IP (optional)
sudo nano /etc/netplan/50-cloud-init.yaml
```

3.2 Static IP Configuration

yaml

```yaml
# /etc/netplan/50-cloud-init.yaml
network:
  version: 2
  wifis:
    wlan0:
      dhcp4: false
      addresses: [192.168.221.227/24]
      gateway4: 192.168.221.1
      nameservers:
        addresses: [8.8.8.8, 1.1.1.1]
      access-points:
        "Darahas's Vivo V23":
          password: "naga@4321"
```

bash

```bash
# Apply network configuration
sudo netplan apply
```

📦 Step 4: System Updates and Dependencies

4.1 System Update

bash

```
# Update package lists
sudo apt update

# Upgrade system packages
sudo apt upgrade -y

# Install essential packages
sudo apt install -y \
    python3-pip \
    python3-dev \
    python3-setuptools \
    python3-tk \
    git \
    wget \
    curl \
    nano \
    htop \
    build-essential
```

4.2 Python Environment Setup

bash

```
# Check Python version
python3 --version  # Should be 3.6+

# Install pip packages
pip3 install --upgrade pip

# Install core Python packages
pip3 install \
    numpy \
    matplotlib \
    requests \
    flask \
    tkinter \
    joblib \
    scipy \
    pandas
```

🤖 Step 5: PyTorch Installation

5.1 PyTorch for Jetson Nano

bash

```bash
# Download PyTorch wheel (pre-built for Jetson)
wget https://nvidia.box.com/shared/static/p57jwntv436lfrd78inwl7iml6p13fzh.whl -O torch-1.8.0-cp36-cp36m-linux_aa

# Install PyTorch
pip3 install torch-1.8.0-cp36-cp36m-linux_aarch64.whl

# Install torchvision
sudo apt install -y libopenblas-base libopenmpi-dev
pip3 install torchvision

# Verify installation
python3 -c "import torch; print(torch.__version__); print(torch.cuda.is_available())"
```

5.2 Additional ML Libraries

```bash
# Install scikit-learn
pip3 install scikit-learn

# Install additional dependencies
pip3 install \
    pillow \
    opencv-python \
    tqdm \
    matplotlib
```

🔌 Step 6: GPIO Configuration

6.1 Install Jetson.GPIO

```bash
# Install Jetson GPIO library
sudo pip3 install Jetson.GPIO

# Add user to gpio group
sudo groupadd -f -r gpio
sudo usermod -a -G gpio $USER

# Set GPIO permissions
sudo cp /opt/nvidia/jetson-io/jetson-io.py /usr/local/bin/
sudo chmod +x /usr/local/bin/jetson-io.py
```

6.2 GPIO Pin Configuration

python

```python
# Test GPIO configuration
import Jetson.GPIO as GPIO

# Set board mode
GPIO.setmode(GPIO.BOARD)

# LED pins from your code
LED_PINS = [33, 32, 31, 29, 18]

# Setup pins as output
for pin in LED_PINS:
    GPIO.setup(pin, GPIO.OUT)
    GPIO.output(pin, GPIO.LOW)

print("GPIO configured successfully!")
GPIO.cleanup()
```

📁 Step 7: Project Directory Setup

7.1 Create Project Structure

bash

```bash
# Create main project directory
mkdir -p /home/jetson/Desktop/pd_processing_system
cd /home/jetson/Desktop/pd_processing_system

# Create subdirectories
mkdir -p {cnn_models,llm_models,scripts,logs,data}

# Create directory structure
pd_processing_system/
├── cnn_models/
│   ├── multiclass_cnn_model
│   ├── label_classes.npy
│   └── multiclass_scaler.pkl
├── llm_models/
│   └── final_parkinsons_model.pth
├── scripts/
│   └── jetson_final_code.py
├── logs/
└── data/
```

7.2 Set Permissions

bash

```
# Set proper permissions
chmod +x /home/jetson/Desktop/pd_processing_system/scripts/
chown -R jetson:jetson /home/jetson/Desktop/pd_processing_system/
```

🧠 Step 8: Model Setup

8.1 Model Directory Configuration

bash

```
# Update paths in your Python script
MODEL_PATHS = {
    'cnn_model': "/home/jetson/Desktop/pd_processing_system/cnn_models/multiclass_cnn_model",
    'label_classes': "/home/jetson/Desktop/pd_processing_system/cnn_models/label_classes.npy",
    'scaler': "/home/jetson/Desktop/pd_processing_system/cnn_models/multiclass_scaler.pkl",
    'llm_model': "/home/jetson/Desktop/pd_processing_system/llm_models/final_parkinsons_model.pth"
}
```

8.2 Model Loading Test

python

```
# Test model loading
import torch
import joblib
import numpy as np

# Test paths
cnn_path = "/home/jetson/Desktop/pd_processing_system/cnn_models/multiclass_cnn_model"
scaler_path = "/home/jetson/Desktop/pd_processing_system/cnn_models/multiclass_scaler.pkl"

# Load and test
try:
    scaler = joblib.load(scaler_path)
    print("✅ Scaler loaded successfully")

    model_state = torch.load(cnn_path, map_location=torch.device('cpu'))
    print("✅ CNN model loaded successfully")

except Exception as e:
    print(f"❌ Error loading models: {e}")
```

⚡ Step 9: System Optimization

9.1 Performance Configuration

```bash
# Enable maximum performance mode
sudo nvpmodel -m 0

# Check current power mode
sudo nvpmodel -q

# Set CPU governor to performance
echo 'performance' | sudo tee /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor

# Increase swap (for 32GB SD card)
sudo fallocate -l 2G /swapfile
sudo chmod 600 /swapfile
sudo mkswap /swapfile
sudo swapon /swapfile

# Make swap permanent
echo '/swapfile none swap sw 0 0' | sudo tee -a /etc/fstab
```

9.2 Memory Optimization

```bash
# Check memory usage
free -h

# Configure system for 4GB RAM
echo 'vm.swappiness=10' | sudo tee -a /etc/sysctl.conf
echo 'vm.vfs_cache_pressure=50' | sudo tee -a /etc/sysctl.conf

# Apply settings
sudo sysctl -p
```

🚀 Step 10: Auto-Start Configuration

10.1 Create Systemd Service

```bash
```

```
# Create service file
sudo nano /etc/systemd/system/pd-processor.service
```

ini

```
[Unit]
Description=PD Auto-Processing Service
After=network.target

[Service]
Type=simple
User=jetson
WorkingDirectory=/home/jetson/Desktop/pd_processing_system/scripts
ExecStart=/usr/bin/python3 jetson_final_code.py
Restart=always
RestartSec=10

[Install]
WantedBy=multi-user.target
```

10.2 Enable Auto-Start

bash

```
# Enable service
sudo systemctl enable pd-processor.service

# Start service
sudo systemctl start pd-processor.service

# Check status
sudo systemctl status pd-processor.service
```

🧪 Step 11: Testing and Verification

11.1 Hardware Test

python

```
# GPIO LED Test Script
import Jetson.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)
led_pins = [33, 32, 31, 29, 18]

for pin in led_pins:
    GPIO.setup(pin, GPIO.OUT)

# Test each LED
for i, pin in enumerate(led_pins):
    print(f"Testing LED {i+1} on pin {pin}")
    GPIO.output(pin, GPIO.HIGH)
    time.sleep(1)
    GPIO.output(pin, GPIO.LOW)

GPIO.cleanup()
print("✅ All LEDs tested successfully!")
```

11.2 Network Test

bash

```
# Test HTTP server
python3 -c "
import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(('0.0.0.0', 8888))
s.listen(1)
print('✅ Port 8888 available')
s.close()
"

# Test network connectivity
ping -c 3 192.168.221.101  # ESP32 IP
```

📊 Step 12: System Monitoring

12.1 Resource Monitoring

bash

```bash
# Install monitoring tools
sudo apt install -y iotop iftop

# Monitor system resources
htop                # CPU and memory
iotop               # Disk I/O
iftop               # Network traffic
nvidia-smi          # GPU usage (if applicable)

# Check disk usage (important for 32GB)
df -h
```

12.2 Storage Management

```bash
# Clean up unnecessary packages
sudo apt autoremove -y
sudo apt autoclean

# Check large files
du -sh /home/jetson/* | sort -hr

# Monitor disk usage
watch -n 5 'df -h /'
```

🔧 Step 13: Troubleshooting

13.1 Common Issues

```bash
# GPU memory issues
echo 'export CUDA_VISIBLE_DEVICES=0' >> ~/.bashrc

# Permission issues
sudo chown -R jetson:jetson /home/jetson/
sudo chmod -R 755 /home/jetson/Desktop/pd_processing_system/

# Network issues
sudo systemctl restart NetworkManager
```

13.2 Log Monitoring

```bash
```

```
# System logs
journalctl -u pd-processor.service -f

# Python application logs
tail -f /home/jetson/Desktop/pd_processing_system/logs/app.log

# GPIO issues
dmesg | grep gpio
```

✅ Step 14: Final Configuration Checklist

Pre-Deployment Verification:

- [ ] ✅ SD card properly flashed and expanded
- [ ] ✅ Network configured (IP: 192.168.221.227)
- [ ] ✅ Python environment with PyTorch installed
- [ ] ✅ Jetson.GPIO library working
- [ ] ✅ All 5 LED pins (33,32,31,29,18) configured
- [ ] ✅ Models loading successfully
- [ ] ✅ HTTP server binding to port 8888
- [ ] ✅ Communication with ESP32 tested
- [ ] ✅ Auto-start service enabled
- [ ] ✅ Performance mode enabled
- [ ] ✅ Sufficient storage available (>5GB free)

System Information:

bash

```
# Check final configuration
echo "=== Jetson Nano Configuration ==="
echo "OS: $(lsb_release -d | cut -f2)"
echo "Kernel: $(uname -r)"
echo "Python: $(python3 --version)"
echo "PyTorch: $(python3 -c 'import torch; print(torch.__version__)')"
echo "GPIO: $(python3 -c 'import Jetson.GPIO; print("Available")')"
echo "Memory: $(free -h | grep Mem | awk '{print $2}')"
echo "Storage: $(df -h / | tail -1 | awk '{print $4}') free"
echo "Network: $(hostname -I)"
echo "================================="
```

🚨 Important Notes for 32GB SD Card:

1. **Storage Management**: With 32GB, monitor disk usage regularly
2. **Swap Configuration**: 2GB swap file configured for better performance
3. **Model Storage**: Keep only essential models to save space
4. **Log Rotation**: Implement log rotation to prevent disk full
5. **Regular Cleanup**: Run `sudo apt autoremove` periodically

Your Jetson Nano is now configured and ready for the PD Auto-Processing System!