

# 1 Thuật toán tăng cường dữ liệu thích ứng

## 1.1 Cơ sở lý thuyết

Trong nhận diện biểu cảm khuôn mặt (FER) dưới điều kiện ánh sáng yếu, chất lượng ảnh thường bị suy giảm do độ sáng thấp. Các kỹ thuật tăng cường dữ liệu cố định như *histogram equalization* hoặc *gamma correction* không tối ưu vì chúng áp dụng đồng nhất cho mọi ảnh. Thuật toán **tăng cường dữ liệu thích ứng** được thiết kế để phân tích mức độ ánh sáng của từng ảnh và áp dụng kỹ thuật tăng cường phù hợp.

Thuật toán dựa trên các cơ sở lý thuyết sau:

- **Phân tích độ sáng:** Độ sáng trung bình ( $\mu$ ) của ảnh grayscale được sử dụng để đánh giá mức độ ánh sáng yếu.
- **Kỹ thuật tăng cường:**
  - *Gamma correction:*  $I_{\text{out}} = I_{\text{in}}^\gamma$ , với  $\gamma < 1$  để tăng độ sáng.
  - *Histogram equalization:* Tăng độ tương phản bằng cách phân bố lại giá trị pixel.
  - *Contrast stretching:* Kéo giãn khoảng giá trị pixel.
- **Tính thích ứng:** Dựa trên  $\mu$  và các ngưỡng  $T_1, T_2$ , thuật toán chọn kỹ thuật tăng cường phù hợp.

## 1.2 Mô tả thuật toán

Thuật toán được trình bày như sau:

## 1.3 Ưu điểm

- **Tính thích ứng:** Tùy chỉnh tăng cường dựa trên đặc trưng ánh sáng của từng ảnh.
- **Đơn giản:** Dễ triển khai, phù hợp với mô hình nhẹ như MobileNetV3.
- **Hiệu quả:** Cải thiện chất lượng ảnh, hỗ trợ nhận diện biểu cảm tốt hơn.

---

**Algorithm 1** Tăng cường dữ liệu thích ứng

---

**Require:** Ảnh đầu vào  $I$ , ngưỡng  $T_1, T_2$ , giá trị gamma  $\gamma_{\text{low}}, \gamma_{\text{mid}}$

**Ensure:** Ảnh tăng cường  $I_{\text{out}}$

```
1: Chuyển  $I$  sang grayscale để tính độ sáng trung bình  $\mu$ 
2: if  $\mu < T_1$  then
3:   Áp dụng gamma correction với  $\gamma = \gamma_{\text{low}}$ 
4:    $I_{\text{out}} \leftarrow \text{GammaCorrection}(I, \gamma_{\text{low}})$ 
5: else if  $T_1 \leq \mu < T_2$  then
6:   Áp dụng gamma correction với  $\gamma = \gamma_{\text{mid}}$ 
7:    $I_{\text{out}} \leftarrow \text{GammaCorrection}(I, \gamma_{\text{mid}})$ 
8: else
9:   Áp dụng contrast stretching
10:   $I_{\text{out}} \leftarrow \text{ContrastStretching}(I)$ 
11: end if
12: return  $I_{\text{out}}$ 
```

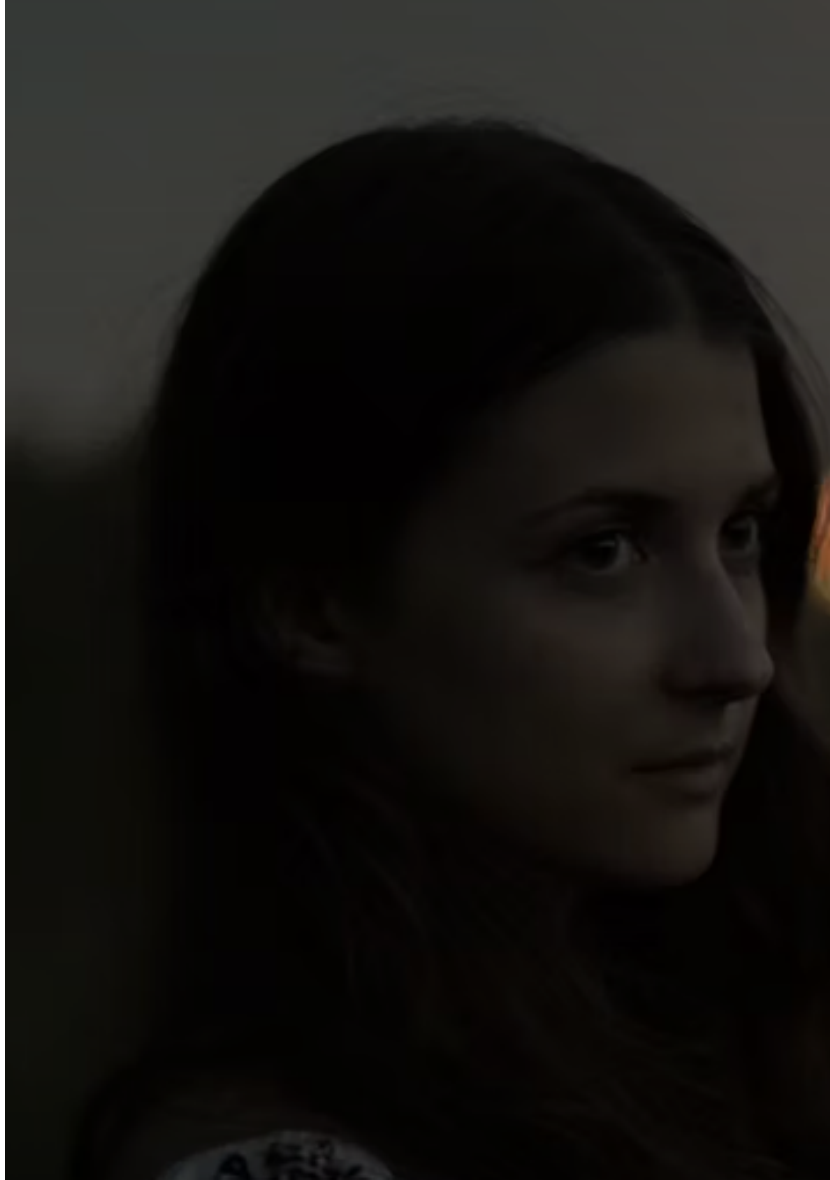
---

## 1.4 Khuyết điểm

- **Phụ thuộc ngưỡng:** Hiệu quả phụ thuộc vào việc chọn  $T_1, T_2$ .
- **Hạn chế với ánh sáng không đồng đều:** Không xử lý tốt vùng tối cục bộ.
- **Không xử lý nhiễu:** Có thể cần thêm bước giảm nhiễu.

## 1.5 Triển khai thuật toán

Thuật toán được triển khai bằng Python với các thư viện OpenCV và NumPy. Dưới đây là một ví dụ minh họa:



Hình 1: Kết quả tăng cường dữ liệu thích ứng trên ảnh ánh sáng yếu.