FINAL CODE(sender)

```
#include "esp_camera.h"
#include <WiFi.h>
#include <HTTPClient.h>

#define CAMERA_MODEL_AI_THINKER
#include "camera_pins.h"

const char *ssid = "MAKERSPACE";
const char *password = "12345678";

const char *ledControllerIP = "http://192.168.0.186";
String lastState = "";  // To avoid sending repeats

void setup() {
 Serial.begin(115200);
 Serial.setDebugOutput(true);
 Serial.println();

 // Connect WiFi
 WiFi.begin(ssid, password);
 Serial.print("Connecting to WiFi");
 while (WiFi.status() != WL_CONNECTED) {
   delay(500);
   Serial.print(".");
 }
 Serial.println("\nConnected! IP address: ");
 Serial.println(WiFi.localIP());

 // Camera
 camera_config_t config;
 config.ledc_channel = LEDC_CHANNEL_0;
 config.ledc_timer = LEDC_TIMER_0;
 config.pin_d0 = Y2_GPIO_NUM;
 config.pin_d1 = Y3_GPIO_NUM;
 config.pin_d2 = Y4_GPIO_NUM;
 config.pin_d3 = Y5_GPIO_NUM;
 config.pin_d4 = Y6_GPIO_NUM;
 config.pin_d5 = Y7_GPIO_NUM;
 config.pin_d6 = Y8_GPIO_NUM;
```

```
 config.pin_d7 = Y9_GPIO_NUM;
 config.pin_xclk = XCLK_GPIO_NUM;
 config.pin_pclk = PCLK_GPIO_NUM;
 config.pin_vsync = VSYNC_GPIO_NUM;
 config.pin_href = HREF_GPIO_NUM;
 config.pin_sccb_sda = SIOD_GPIO_NUM;
 config.pin_sccb_scl = SIOC_GPIO_NUM;
 config.pin_pwdn = PWDN_GPIO_NUM;
 config.pin_reset = RESET_GPIO_NUM;
 config.xclk_freq_hz = 20000000;
 config.frame_size = FRAMESIZE_QVGA;
 config.pixel_format = PIXFORMAT_GRAYSCALE;
 config.fb_location = CAMERA_FB_IN_PSRAM;
 config.jpeg_quality = 10;
 config.fb_count = 1;
 config.grab_mode = CAMERA_GRAB_LATEST;

 // Initialize
 esp_err_t err = esp_camera_init(&config);
 if (err != ESP_OK) {
   Serial.printf("Camera init failed with error 0x%x", err);
   return;
 }

 Serial.println("Camera initialized!");
}

void loop() {
 camera_fb_t *fb = esp_camera_fb_get();
 if (!fb) {
   Serial.println("Camera capture failed");
   delay(2000);
   return;
 }

 uint32_t totalBrightness = 0;
 int pixelCount = fb->len;

 for (int i = 0; i < pixelCount; i++) {
   totalBrightness += fb->buf[i];
 }
```

```
 int avgBrightness = totalBrightness / pixelCount;
 Serial.printf("Average brightness: %d\n", avgBrightness);

 String currentState;

 if (avgBrightness < 120) {
   currentState = "dark";
   Serial.println("Scene is DARK");
 } else if (avgBrightness > 130) {
   currentState = "white";
   Serial.println("Scene is BRIGHT");
 }

 // Only send request if state changed
 if (currentState != lastState) {
   if (currentState == "dark") {
     sendCommand("/led/on");
   } else if (currentState == "white") {
     sendCommand("/led/off");
   }
   lastState = currentState;
 }

 esp_camera_fb_return(fb);
 delay(2000);
}

void sendCommand(const char *endpoint) {
 if (WiFi.status() == WL_CONNECTED) {
   HTTPClient http;
   String url = String(ledControllerIP) + endpoint;
   http.begin(url);
   int httpCode = http.GET();
   if (httpCode > 0) {
     Serial.printf("Sent request to %s — Response code: %d\n", url.c_str(), httpCode);
   } else {
     Serial.printf("Failed to send request to %s\n", url.c_str());
   }
   http.end();
 } else {
   Serial.println("WiFi not connected");
 }
```

```
}
```

FINAL CODE(receiver)

```
#include <WiFi.h>
#include <WebServer.h>
#include <Adafruit_NeoPixel.h>
#include <ESPmDNS.h>

const char* ssid = "MAKERSPACE";
const char* password = "12345678";

#define LED_PIN 25         // Pin connected to LED strip
#define NUM_LEDS 10        // Number of LEDs in strip

Adafruit_NeoPixel strip(NUM_LEDS, LED_PIN, NEO_GRB + NEO_KHZ800);
WebServer server(80);

void handleLedOn() {
 for (int i = 0; i < NUM_LEDS; i++) {
   strip.setPixelColor(i, strip.Color(255, 105, 180)); // pink
 }
 strip.show();
 server.send(200, "text/plain", "LEDs ON");
}

void handleLedOff() {
 strip.clear();
 strip.show();
 server.send(200, "text/plain", "LEDs OFF");
}

void handleRoot() {
 String html = "<h1>ESP32 LED Control</h1>"
               "<p><a href=\"/led/on\">Turn LEDs ON</a></p>"
               "<p><a href=\"/led/off\">Turn LEDs OFF</a></p>";
 server.send(200, "text/html", html);
}

void setup() {
 Serial.begin(115200);
 strip.begin();
 strip.show();  // Initialize all pixels to off
```

```
 WiFi.begin(ssid, password);
 Serial.print("Connecting to WiFi");
 while (WiFi.status() != WL_CONNECTED) {
   delay(500);
   Serial.print(".");
 }

 Serial.println("\nConnected! IP address: ");
 Serial.println(WiFi.localIP());

 if (MDNS.begin("esp32")) {
   Serial.println("mDNS responder started — try http://esp32.local/");
 }

 server.on("/", handleRoot);
 server.on("/led/on", handleLedOn);
 server.on("/led/off", handleLedOff);

 server.begin();
 Serial.println("HTTP server started");
}

void loop() {
 server.handleClient();
}
```