S - Aplikace strojového učení na vestavných platformách

Pavel Stepanov
xstepa77

December 15, 2024

Contents

1 Úvod

1.1 Motivace

Motivací tohoto projektu je demonstrovat možnosti **strojového učení** (ML) na zařízeních s **omezenými výpočetními prostředky**. Projekt se zaměřuje na realizaci systému pro **rozpoznávání barev** na platformě **ESP32** s využitím **TensorFlow Lite for Microcontrollers (TFLM)**.

Cílem je vytvořit aplikaci schopnou:

- Spolehlivě rozpoznávat barvy (Red, Green, Blue).
- Efektivně využívat zdroje platformy ESP32 (Flash a RAM).
- Poskytovat rychlou inferenci vhodnou pro provoz v reálném čase.

1.2 Cíle projektu

Hlavní cíle projektu:

1. Návrh a trénování neuronové sítě pro rozpoznávání barev na **PC** (Python + TensorFlow).
2. Optimalizace modelu a převod do formátu **TensorFlow Lite**.
3. Implementace kódu pro čtení hodnot z **barevného senzoru** a jejich předání neuronové síti na ESP32.
4. Měření výkonu: čas inference, využití RAM a Flash.
5. Testování systému na reálných datech k ověření přesnosti.

1.3 Očekávané výsledky

Očekává se, že projekt:

- Umožní rozpoznávat barvy s vysokou přesností i na omezeném hardwaru.
- Ukáže efektivitu neuronových sítí na platformě ESP32.
- Poskytne měřitelné výsledky (přesnost, čas inference, využití paměti).

Projekt je příkladem toho, jak lze moderní technologie ML efektivně využít k řešení úloh na zařízeních s nízkým výkonem, jako je platforma ESP32.

2 Rozbor a návrh řešení

2.1 Použité komponenty a jejich propojení

Vestavná platforma: Pro realizaci projektu bude použita deska **LilyGO TTGO Mini32 ESP32-WROVER-B**., která nabízí:

- Dvoujádrový procesor s frekvencí až 240 MHz.
- Integrovaný Wi-Fi a Bluetooth modul.
- 4 MB Flash a 520 KB RAM.

Barevný senzor: Bude využit **TCS3200**, převádějící intenzitu světla na výstupní frekvenci. Propojení s ESP32 bude realizováno pomocí následujícího pinoutu (viz Obr. 1):

Pin senzoru	**Funkce**	**Pin ESP32**	**Poznámka**
S0	Frekvenční dělič	GPIO25	Nastavení děliče
S1	Frekvenční dělič	GPIO26	Nastavení děliče
S2	Barevný filtr	GPIO27	Nastavení filtru
S3	Barevný filtr	GPIO14	Nastavení filtru
OUT	Výstupní signál	GPIO34	Intenzita světla
VCC	Napájení	3.3V	Zdroj napětí
GND	Zem	GND	Společná zem

Table 1: Propojení senzoru **TCS3200** s ESP32

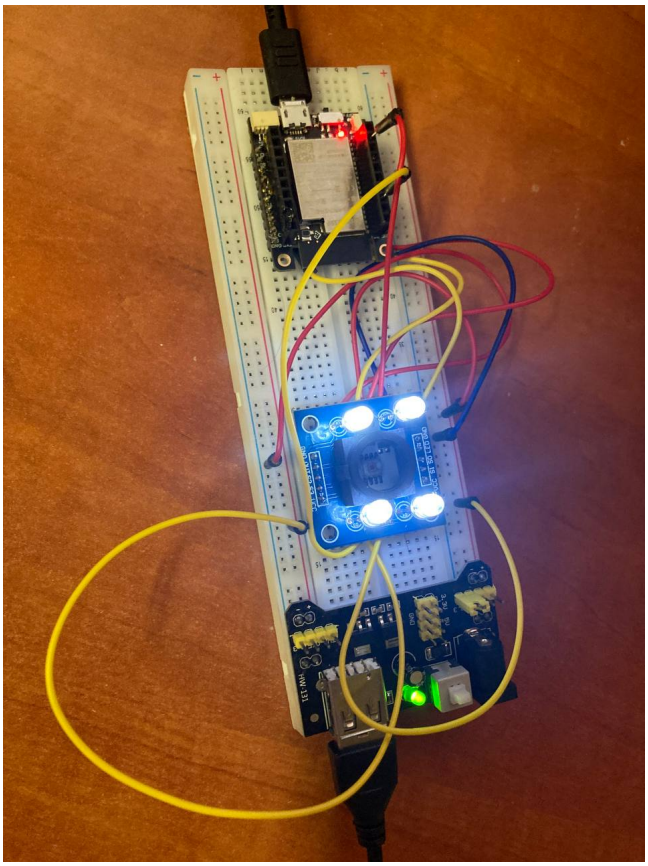

Figure 1: ESP32 s modulem senzoru barev a RGB na nepájivém poli.

Propojení bude provedeno propojovacími vodiči přes nepájivé pole.

2.2 Popis řešeného problému a metodologie

Cílem projektu je vytvořit aplikaci pro **rozpoznávání barev** na platformě ESP32 s použitím **strojového učení**. Plánované kroky:

1. **Sběr dat:** Měření RGB hodnot ze senzoru **TCS3200** v různých podmínkách a jejich ukládání do CSV.
2. **Příprava dat:** Normalizace a přidání příznaků, např. **R/G ratio**, **R - G**, **Dominant G**.
3. **Trénování modelu:** Vytvoření a optimalizace neuronové sítě (TensorFlow/Keras) s následnou konverzí na **TensorFlow Lite for Microcontrollers**.

4. **Nasazení na ESP32:** Nahrání optimalizovaného modelu a jeho integrace s kódem v **Arduino IDE**.

2.3 Struktura řešení

Konečné řešení zahrnuje:

- **Kód v ESP-IDF 5.3** (C++):
  - Inicializace senzoru a čtení RGB hodnot.
  - Výpočet příznaků.
  - Predikce výsledků neuronovou sítí.
- **Model neuronové sítě:** Struktura 8-8-4, optimalizovaná pro ESP32.
- **Knihovna esp-tflite-micro (verze 1.3.2):** Pro inferenci.

2.4 Popis řetězce strojového učení

Plánovaný proces zahrnuje:

1. **Sběr dat:** Hodnoty RGB budou senzoricky měřeny a ukládány do CSV.
2. **Příprava dat:** Data budou normalizována a rozšířena o příznaky pomocí Pythonu.
3. **Trénování modelu:**
   - Použitá knihovna: TensorFlow/Keras.
   - Struktura modelu: 8 vstupních neuronů, 8 skrytých, 4 výstupní (Red, Green, Blue, None).
   - Optimalizace: Minimalizace velikosti a testování na ESP32.
4. **Nasazení:** Model bude integrován do kódu a otestován.

2.5 Použité zdroje a nástroje

Hardwarové komponenty:

- LilyGO TTGO Mini32 ESP32-WROVER-B.
- Barevný senzor **TCS3200**.
- Breadboard.

Softwarové nástroje:

- **ESP-IDF 5.3** a **VS Code**.
- **TensorFlow/Keras**.
- **esp-tflite-micro (verze 1.3.2)**.

Řešení kombinuje hardwarové a softwarové aspekty s důrazem na optimalizaci paměti a inference.

3 Vlastní řešení

3.1 Použitý senzor a jeho omezení

Pro měření hodnot RGB byl zvolen barevný senzor **TCS3200**, který obsahuje fotodiodovou matici a čtyři typy filtrů: červený (*Red*), zelený (*Green*), modrý (*Blue*) a průhledný (*Clear*). Spektrální citlivost senzoru je zobrazena na Obr. 2.

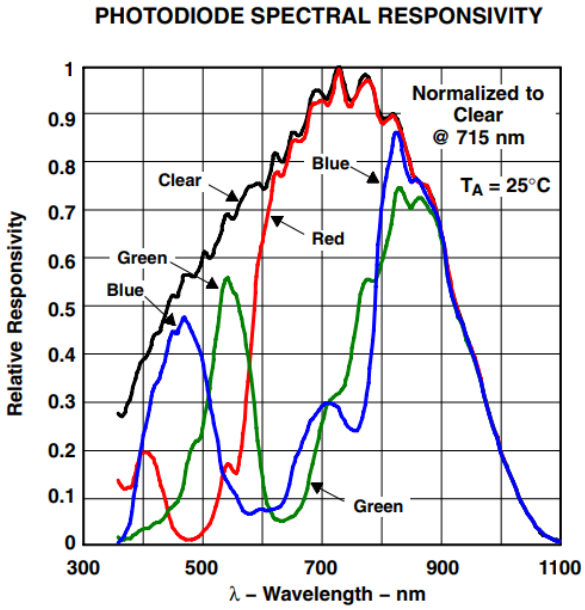

Figure 2: Spektrální citlivost fotodiody senzoru **TCS3200**

Z grafu vyplývá, že každá složka reaguje na různé vlnové délky:

- Modrá složka: citlivost v rozsahu 450–500 nm.
- Zelená složka: vrchol citlivosti kolem 550 nm.
- Červená složka: maximální citlivost mezi 600–700 nm.

Při praktickém použití senzoru se objevily následující problémy:

1. Hodnoty frekvencí byly ovlivněny intenzitou okolního osvětlení.
2. Rozsahy pro červenou a modrou složku se často překrývaly, což vedlo k nejednoznačnosti.
3. Nižší citlivost zelené složky komplikovala její identifikaci.

—

3.2 Zpracování dat a extrakce příznaků

Pro odstranění vlivu zmíněných problémů byla data zpracována následujícím způsobem:

- **Normalizace frekvencí:** Hodnoty byly převedeny na rozsah 0–1:

$$\text{Hodnota}_{\text{norm}} = \frac{\text{Frekvence}_{\text{naměřená}}}{400.0}.$$

- **Výpočet příznaků:**
  - Poměr R/G: $RG\_ratio = \frac{R}{G+10^{-6}}$,
  - Rozdíl R - G: $R\_minus\_G = R - G$,
  - Dominantní G: Hodnota zelené složky.

Naměřená data byla doplněna o **synteticky generovaná data** pro zvýšení jejich objemu a rovnoměrnosti distribuce mezi třídami. Každé původní měření bylo **narušeno gaussovským šumem** s nízkou odchylkou, což zajistilo robustnost modelu vůči senzorovým odchylkám a různým podmínkám osvětlení. Celkem bylo připraveno **5000 datových vstupů**, které reprezentují třídy *Red*, *Green*, *Blue* a *None*.

—

3.3 Trénování a optimalizace modelu

V první fázi byl vytvořen model neuronové sítě s architekturou **64-64-4** (dvě skryté vrstvy po 64 neuronech a čtyři výstupní neurony). Tato neredukovaná varianta poskytovala přesnost mírně nad 99 %, ale měla velké nároky na paměť a byla nevhodná pro nasazení na zařízení s omezenými zdroji.

Optimalizací modelu do struktury **8-8-4** (dvě skryté vrstvy po 8 neuronech) jsme dosáhli následujících výsledků:

- Přesnost modelu zůstala mírně nad 99 % na trénovacích datech.
- Velikost výsledného modelu se snížila **desetkrát**, což umožnilo jeho efektivní běh na platformě ESP32.
- Inferenční čas byl minimalizován na **300 ms** na jeden cyklus zpracování.

Na Obr. 3 a Obr. 4 jsou znázorněny grafy trénování pro neredukovaný a optimalizovaný model.

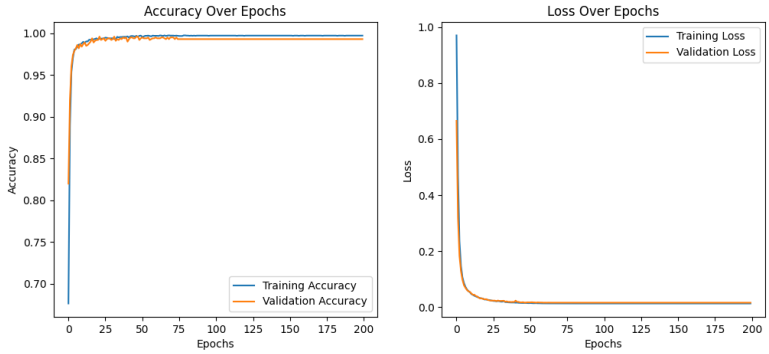

Figure 3: Trénování původního modelu 64-64-4

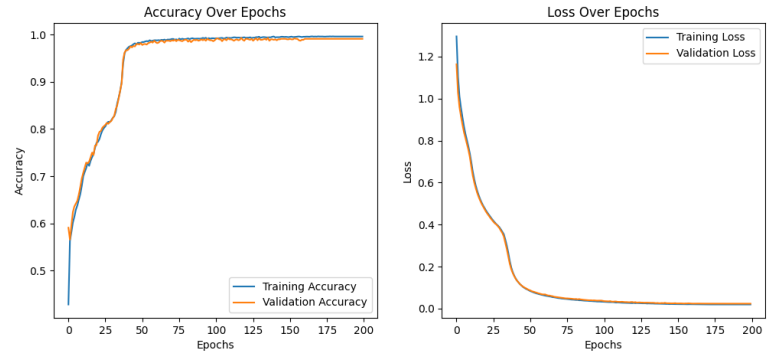

Figure 4: Trénování optimalizovaného modelu 8-8-4

3.4 Implementace na platformě ESP32

Optimalizovaný model byl převeden do formátu **TensorFlow Lite for Microcontrollers** a nasazen na platformě **ESP32** pomocí frameworku **ESP-IDF**. Kroky implementace zahrnovaly:

1. **Konfigurace GPIO:** Pinout senzoru **TCS3200** byl nastaven pomocí funkcí ESP-IDF.
2. **Zpracování dat:** Naměřené frekvence byly znormalizovány a rozšířeny o odvozené příznaky.
3. **Inferenční běh modelu:** Data byla předána neuronové síti pomocí knihovny `TensorFlow Lite Micro`.

4. **Měření výkonu:** Byla sledována rychlost inference a využití paměti.

Parametr	**Hodnota**
Volná paměť (heap)	290736 B
Čas na jeden cyklus	300.74 ms
Velikost modelu	17 kB

Table 2: Přehled výkonu a paměťových nároků optimalizované implementace

—

4 Závěrečné zhodnocení

Zde se můžete podívat na princip fungování projektu na reálných experimentech:
https://drive.google.com/file/d/1CD42iendyMvMEkGteHB_Ra5dmoi5mz6o/view?usp=drive_link
V rámci projektu byl úspěšně realizován systém pro detekci barev na platformě ESP32 pomocí čidla **TCS3200** a neuronové sítě běžící v TensorFlow Lite Micro. Byly splněny všechny stanovené cíle:

- Vytvoření datové sady založené na reálných a simulovaných hodnotách z RGB senzorů.
- Trénování a optimalizace neuronové sítě pro efektivní inferenci na vestavné platformě.
- Implementace aplikace na platformě ESP32 s minimálními požadavky na paměť.

Z výsledků testování vyplynulo, že systém dosahuje přesnosti detekce barev více než 99 %, a to i přes omezené výpočetní prostředky. Zhodnocení vlastností implementovaného systému zahrnuje:

- **Paměťová efektivita:** Model využívá pouze 17 kB paměti díky optimalizaci neuronové sítě.
- **Rychlost inferencí:** Čas detekce jedné barvy činí přibližně 300 ms.
- **Přesnost:** Bylo dosaženo vysoké přesnosti díky vhodně navržené datové sadě a optimalizovanému modelu.

Výsledky projektu ukázaly, že strojové učení lze efektivně implementovat i na zařízeních s omezenými výpočetními prostředky, jako je ESP32.

5 Použité informační zdroje

1. TCS3200 DataSheet: https://www.laskakit.cz/user/related_files/taos-tcs3200-datasheet.pdf
2. LilyGO ESP32 DataSheet: https://www.laskakit.cz/user/related_files/wch-jiangsu-qin-heng-ch9102f.pdf
3. Můj Model trainer and collected data: https://github.com/espatie1/Training_Model.
4. Github TensorFlow Lite Micro https://github.com/tensorflow/tflite-micro.