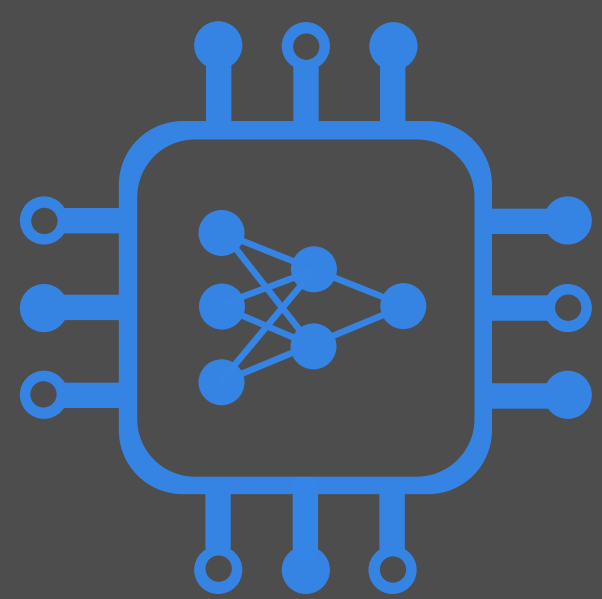
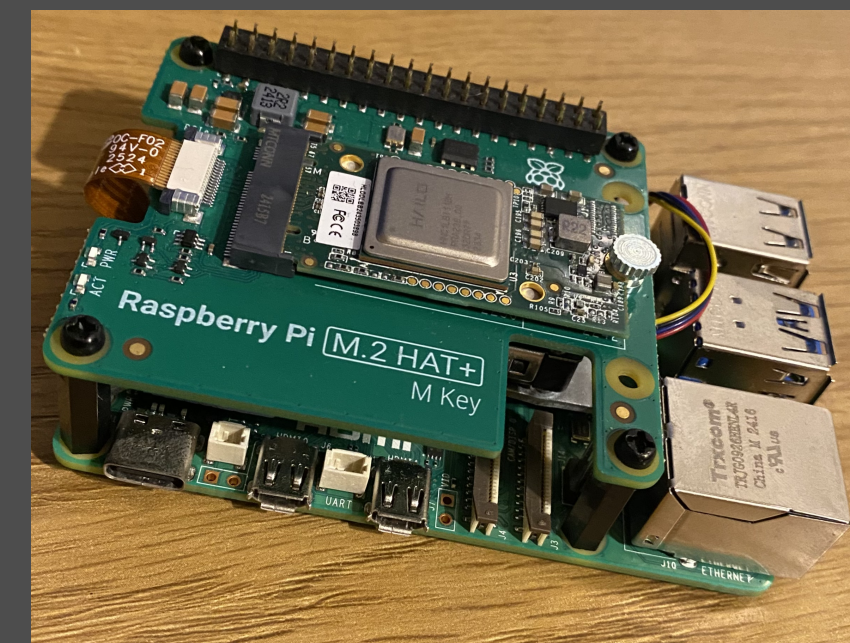




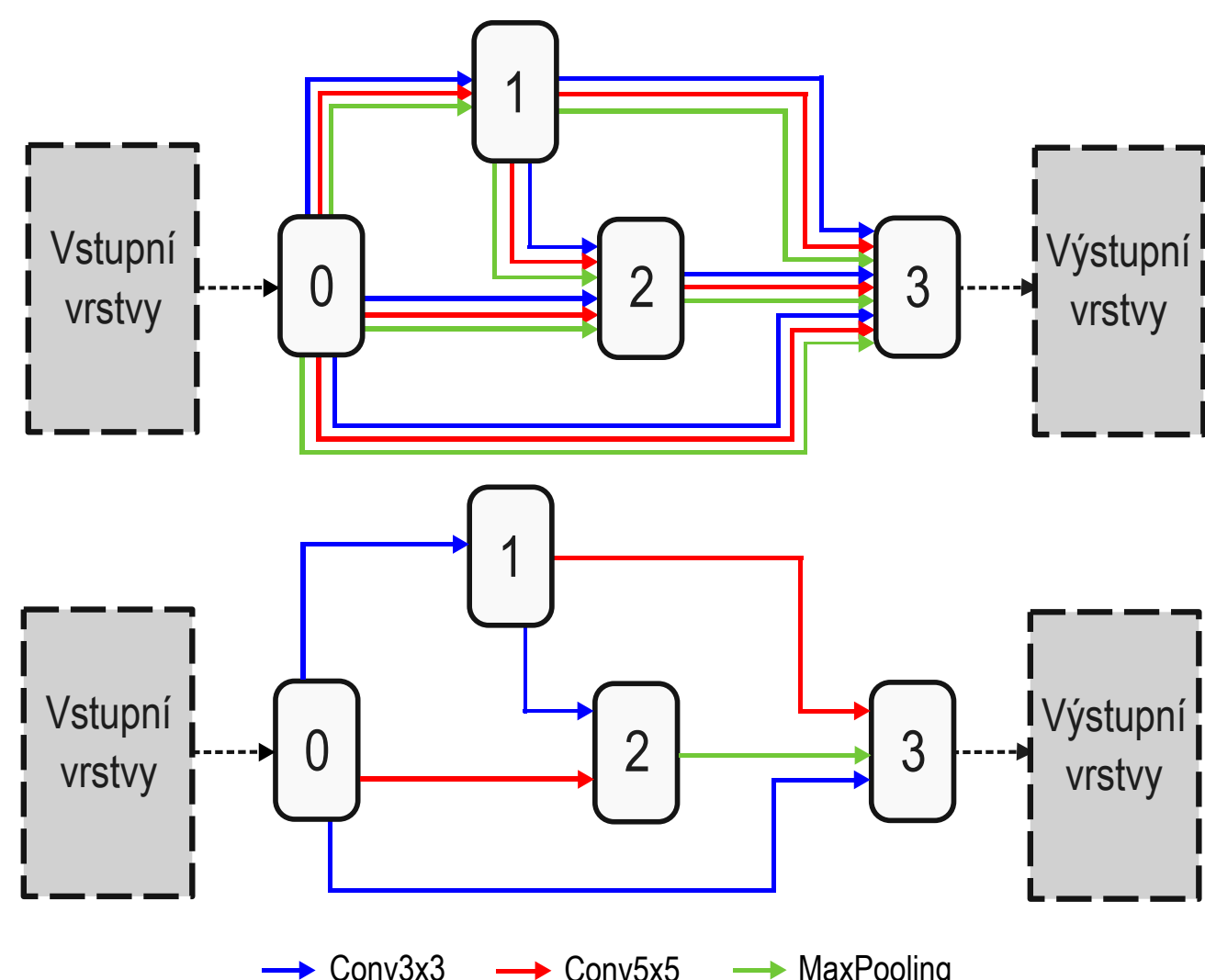
# Automatizovaný návrh neuronových sítí s ohledem na latenci pro edge akcelerátory



## Motivace

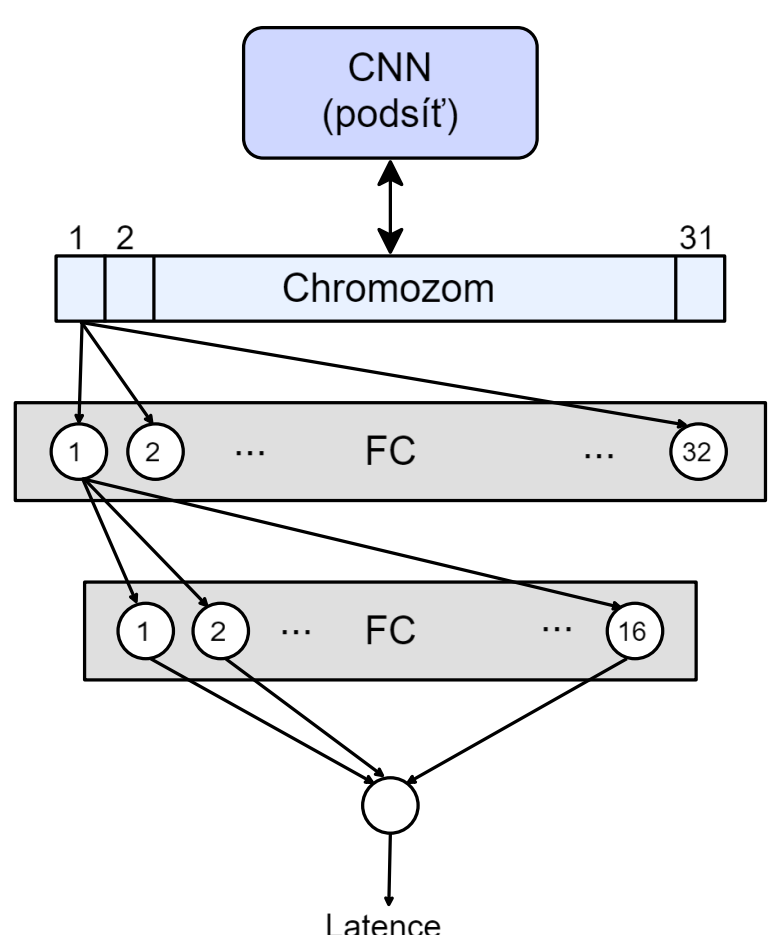
Moderní vestavěné systémy, jako jsou zařízení pro autonomní řízení či chytré kamery, vyžadují rychlé a přesné zpracování obrazu. Optimalizace architektur přímo s ohledem na latenci umožňuje efektivní nasazení v reálném čase.

## Supersít'

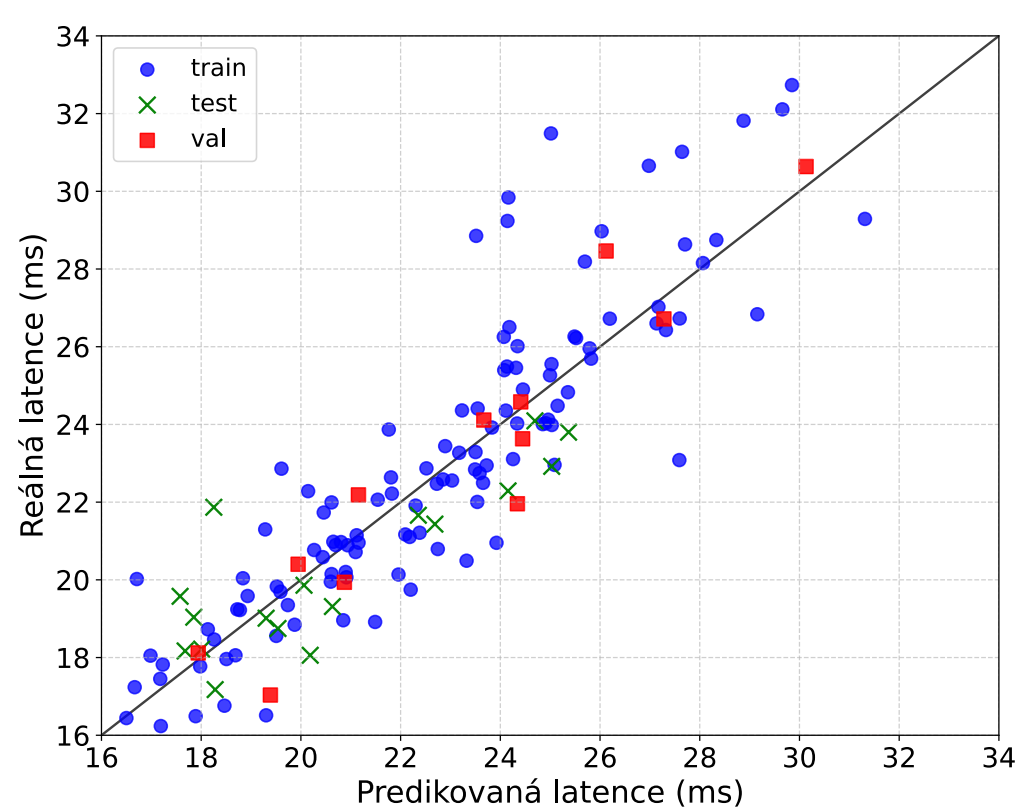


Obrázek 1: Příklad zjednodušené super-sítě (nahore) a jedné z ní odvozené podsítě (dole). Uzly označují feature maps.

## Prediktor latence



Obrázek 2: Architektura modelu prediktoru latence. Jako vstup přijímá konfiguraci architektury podsítě (chromozom - zakódování v průběhu evoluce)



Obrázek 4 znázorňuje korelaci mezi predikovanou a skutečně naměřenou latencí. Tyto výsledky potvrzují, že navržený prediktor poskytuje dostatečně přesné odhady pro potřeby metody NAS.

## Řešení

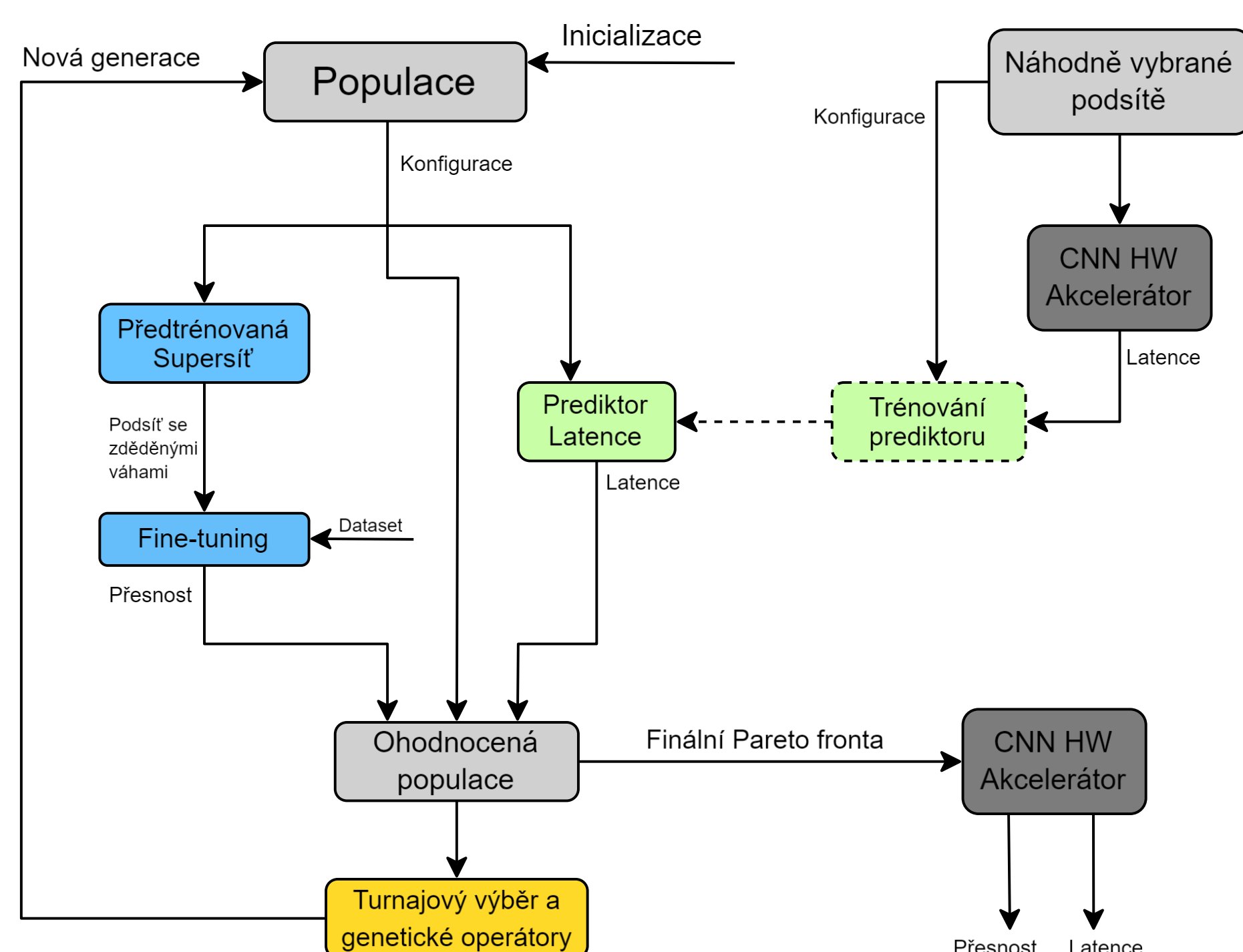
Navrhujeme evoluční Neural Architecture Search (NAS) metodu optimalizující přesnost a latenci CNNs pro akcelerátor Hailo-8L. Metoda kombinuje:

- **Supersít'** - předtrénovaná síť umožňující rychlé vyhodnocení přesnosti podsítí bez potřeby plného tréninku
- **Prediktor latence** - malá neuronová síť rychle odhadující latenci místo nákladného měření na hardwaru
- **NSGA-II** algoritmus - vícekritériální genetická optimalizace hledající nejlepší kompromisy mezi přesností a latencí.

Pro další urychlení využíváme:

- Zrychlený fine-tuning podsítí (1 epocha trénink celé podsítě, další jen poslední vrstva).
- Redukci trénovacích dat během evoluce.

## Schéma algoritmu



Obrázek 3: Schéma znázorňuje průběh celého návrhu, výběr kandidátních architektur pomocí NSGA-II, jejich rychlé vyhodnocení pomocí supersítě a prediktoru latence a následnou validaci nejlepších modelů na reálném akcelerátoru.

## Nastavení experimentů

NSGA-II:

Velikost populace: 28

Pravděpodobnost mutace: 0.35

Počet generací: 50

Pravděpodobnost křížení: 0.9

Trvání jednoho běhu NSGA-II: cca 20.4 hod

## Výsledky

CIFAR-10

| CNN              | Přesnost po NAS (%) | Přesnost (%) | Predikovaná latence (ms) | Reálná latence (ms) |
|------------------|---------------------|--------------|--------------------------|---------------------|
| SCF10-1          | 85.35               | 91.50        | 15.21                    | 16.10               |
| SCF10-2          | 87.41               | 92.41        | 18.35                    | 17.43               |
| SCF10-3          | 89.11               | 93.26        | 25.39                    | 25.44               |
| VGG16-bn         | -                   | 94.16        | -                        | 14.34               |
| MobileNetV2x 1_4 | -                   | 94.22        | -                        | 6.37                |
| RepVGG-a2        | -                   | 94.98        | -                        | 25.99               |

Tabulka 1

CIFAR-100

| CNN              | Přesnost po NAS (%) | Přesnost (%) | Predikovaná latence (ms) | Reálná latence (ms) |
|------------------|---------------------|--------------|--------------------------|---------------------|
| SCF100-1         | 65.61               | 77.97        | 23.78                    | 25.00               |
| SCF100-2         | 63.29               | 77.32        | 20.62                    | 20.52               |
| SCF100-3         | 66.26               | 78.34        | 28.53                    | 28.45               |
| VGG16-bn         | -                   | 74.00        | -                        | 16.74               |
| MobileNetV2x 1_4 | -                   | 75.98        | -                        | 5.90                |
| RepVGG-a2        | -                   | 77.18        | -                        | 27.15               |

Tabulka 2

Pro CIFAR-10 metoda dokáže rychle najít různé architektury s dobrým kompromisem mezi přesností a latencí. U složitějšího CIFAR-100 navržené sítě po krátkém dotrénování překonaly běžně používané modely v přesnosti, za cenu mírně vyšší latence.

Použitá supersít':

Dilin Wang, Chengyue Gong, Meng Li, Qiang Liu, and Vikas Chandra. Alphanet: Improved training of supernets with alpha-divergence.

<https://github.com/facebookresearch/AlphaNet>