



## **Středoškolská technika 2025**

**Setkání a prezentace prací středoškolských studentů na ČVUT**

# **FUNGI GPT - AI KLASIFIKÁTOR KOŽNÍCH HOUBOVÝCH ONEMOCNĚNÍ**

**Claudia Haová**

Gymnázium prof. Jana Patočky, Praha 1, Jindřišská 36  
Jindřišská 966/36, Praha 1 – Nové Město

# 1. Úvod

Kožní onemocnění patří mezi nejčastější zdravotní problémy na světě. Jsou zpravidla snadno viditelná pouhým okem, což z nich činí ideální vzorky pro klasifikaci na základě obrazových dat. V praxi může lékař stanovit diagnózu během několika minut a umělé inteligence dokáže proces výrazně zrychlit a zjednodušit.

V posledních letech se umělá inteligence začala uplatňovat právě v oblasti dermatologie, kde slouží jako nástroj pro přesnější a rychlejší diagnostiku kožních onemocnění. Některé modely schopné rozeznávat kožní problémy už existují, a jejich další vývoj má potenciál významně přispět ke zlepšení zdravotní péče a dostupnosti diagnostiky pro širší veřejnost.

## 2. Cíl práce

Hlavním cílem je vytvořit model umělé inteligence, který bude schopen rozpoznávat a klasifikovat vybrané typy kožních onemocnění na základě obrazových dat. Tento úkol vyžaduje navržení vhodné architektury neuronové sítě, její trénink na dostupném datasetu a následné testování, které prověří její spolehlivost a praktické využití.

Sekundárním cílem je získat hlubší porozumění principům umělé inteligence, především strojovému učení a neuronovým sítím. V rámci práce se klade důraz také na osvojení základů programování v jazyce Python a seznámení se s nástroji jako je knihovna TensorFlow, která je běžně využívána při vývoji modelů pro zpracování obrazových dat.

## 3. Teoretická část

### 3.1 Kožní plísně

Kožní plísně jsou skupinou infekcí způsobených různými druhy hub, které napadají kůži a způsobují typické změny.

#### 3.1.1 Tinea corporis

Tinea corporis je kožní plíseň postihující zejména tělo, projevující se svědivými kruhovitými skvrnami s ostrými okraji.<sup>1</sup>

#### 3.1.2 Tinea versicolor

Tinea versicolor je chronická plísněvá infekce, která způsobuje změny barvy kůže, často světlé nebo tmavé skvrny, nejvíc na trupu.<sup>2</sup>

#### 3.1.3 Tinea pedis

Tinea pedis, známá také jako atletická noha, je plíseň postihující pokožku mezi prsty na nohou a způsobuje svědění, olupování a zarudnutí.<sup>3</sup>

#### 3.1.4 Candidiasis

Candidiasis je kožní infekce způsobená kvasinkami rodu *Candida*, která může postihnout různé části těla, často vlhké oblasti, a projevuje se zarudnutím, svěděním a podrážděním.<sup>4</sup>

### 3.2 Pojmy

#### 3.2.1 Image classification

Image classification je úloha v oblasti umělé inteligence, která se zaměřuje na rozpoznávání a zařazování obrázků do předem definovaných kategorií.<sup>5</sup>

---

<sup>1</sup> Srov. Solen.cz, 2022, *Tinea corporis u dvouleté dívky*,  
<https://www.solen.cz/pdfs/der/2022/02/10.pdf>

<sup>2</sup> Srov. Mayo Clinic, 2024, *Tinea versicolor*,  
<https://www.mayoclinic.org/diseases-conditions/tinea-versicolor/symptoms-causes/syc-20378385>

<sup>3</sup> Srov. Mayo Clinic, 2023, *Athlete's foot (tinea pedis)*,  
<https://www.mayoclinic.org/diseases-conditions/athletes-foot/symptoms-causes>

<sup>4</sup> Srov. CDC, 2024, *Candidiasis*,  
<https://www.cdc.gov/fungal/diseases/candidiasis/index.html>

<sup>5</sup> Srov. SuperAnnotate, 2023, *What is image classification? Basics you need to know*,  
<https://www.superannotate.com/blog/image-classification-basics>

### 3.2.2 Python

Python je programovací jazyk, který je díky své jednoduchosti a čitelnosti velmi populární v oblasti datové vědy, strojového učení a vývoje umělé inteligence.<sup>6</sup> Nabízí rozsáhlou knihovnu nástrojů a frameworků vhodných pro vývoj a trénink modelů.

### 3.2.3 Jupyter Notebook

Jupyter Notebook je nástroj, který umožňuje psát, spouštět a dokumentovat kód přímo v internetovém prohlížeči buď lokálně nebo na platformách jako Google Colab.<sup>7</sup> Je velmi oblíbený ve výzkumu i výuce, protože kombinuje text, obrázky, matematické vzorce i funkční kód v jednom přehledném prostředí.

Začátečníkům výrazně usnadňuje práci s programováním, jednotlivé části kódu (tzv. buňky) lze spouštět postupně, což pomáhá lépe pochopit, který krok je proveden.

### 3.2.4 TensorFlow

TensorFlow je open-source knihovna obsahující funkce, třídy a nástroje vyvinutá společností Google, která umožňuje vytvářet a trénovat modely strojového učení a neuronových sítí. Je optimalizovaná pro práci s rozsáhlými daty a často se používá v projektech využívajících hluboké učení.<sup>8</sup>

### 3.2.5 Keras

Keras je nadstavba knihovny TensorFlow, která výrazně zjednodušuje tvorbu neuronových sítí. Umožňuje rychlé prototypování a nabízí intuitivní rozhraní pro návrh a trénování modelů.<sup>9</sup> Místo zabývání se komplexních back-end detailů v TensorFlow je možno snadnější sestavení AI modelu pomocí jednodušší syntaxe jazyka Python.

### 3.2.6 ImageDataGenerator

ImageDataGenerator je nástroj z TensorFlow/Keras, který slouží k načítání, předzpracování a augmentaci obrazových dat. Umožňuje načítat obrázky ze složek uspořádaných podle tříd a provádí augmentaci, náhodné úpravy obrázků.<sup>10</sup>

---

<sup>6</sup> Srov. Python Software Foundation, 2024, *What is Python? Executive Summary*, <https://www.python.org/doc/essays/blurb/>

<sup>7</sup> Srov. Jupyter.org, 2024, *Jupyter Notebook*, <https://jupyter.org/>

<sup>8</sup> Srov. TensorFlow, 2024, *Introduction to TensorFlow*, <https://www.tensorflow.org/learn>

<sup>9</sup> Srov. Keras, 2024, *Keras: Deep Learning for humans*, <https://keras.io/>

<sup>10</sup> Srov. TensorFlow, 2024, *tf.keras.preprocessing.image.ImageDataGenerator*, [https://www.tensorflow.org/api\\_docs/python/tf/keras/preprocessing/image/ImageDataGenerator](https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator)

### 3.2.7 Convolutional Neural Network (CNN)

CNN je typ umělé neuronové sítě (program), který se používá k rozpoznávání obrázků. Vznikl podle principu fungování neuronů, kde místo nervových buněk má „umělé neurony“, které zpracovávají data a učí se z předloh podobně jako lidský mozek.<sup>11</sup>

## 3.3 Biologická data v AI

Biologická data v AI představují informace z biologických systémů, například snímky kůže, genetické nebo laboratorní výsledky, které se využívají k trénování modelů. Problémem však je, že sběr těchto dat je často velmi náročný a komplikovaný, zejména pokud není jasně definovaný a legitimní účel jejich použití. Mnohé instituce a jednotlivci proto nesouhlasí s komercializací AI modelů, které využívají jejich data k trénování bez dostatečné transparentnosti nebo souhlasu. Tento nedostatek jasných pravidel a etických standardů může vést k problémům s ochranou soukromí, důvěrou veřejnosti a spravedlivým přístupem k datům, což je třeba brát v úvahu při vývoji a nasazení AI systémů v medicíně.

## 4. Materiály a metodika

Projekt probíhal od února do května 2025 a byl rozdělen do několika hlavních etap. K realizaci byl využit notebook s internetovým připojením. V úvodní fázi bylo důležité si osvojit základy programování v jazyce Python a porozumět principům umělé inteligence, především v oblasti neuronových sítí.

Součástí přípravy bylo také vytvoření vhodného vývojového prostředí. Pro tento projekt byla zvolena platforma Google Colab, která funguje na bázi Jupyter Notebooku a umožňuje pohodlné psaní a spuštění kódu v prohlížeči. Kombinace Pythonu s knihovnou TensorFlow a jejím rozšířením (nadbavbou) Keras poskytla silné zázemí pro vývoj a trénování neuronové sítě.

Dalším krokem byla práce s daty, bylo potřeba nasbírat co nejvhodnější fotografie kožních plísňí, data vyčistit a upravit (například sjednotit rozměry obrázků, normalizovat hodnoty pixelů nebo rozdělit soubor na trénovací a testovací sady).

Pro strukturu modelu byla zvolena Konvoluční neuronová síť (CNN), protože se osvědčila jako efektivní při zpracování vizuálních dat. Architektura sítě byla navržena a upravena dle potřeby na základě průběžného testování.

V průběhu práce probíhaly také případné konzultace s odborníky na umělou inteligenci, bioinformatiku a počítačové vědy. Závěrečná část projektu se zaměřila na testování modelu,

---

<sup>11</sup> Srov. IBM, 2024, *What are Convolutional Neural Networks?*, <https://www.ibm.com/think/topics/convolutional-neural-networksjupyter-notebook.readthedocs.io+3>

ladění datasetu (např. komprese složek), a debugging (hledání a opravování chyb v počítačovém programu).

## 5. Praktická část

### 5.1 Příprava

V úvodní části projektu bylo potřeba se naučit základy programování v jazyce Python a pochopit, jak funguje umělá inteligence. K tomu pomohly online kurzy a edukační videa, například jedenáctihodinové „crash course“ nebo začátečnický kurz počítačové vědy na Khan Academy. Díky těmto materiálům jsem získala základní přehled o tom, jak neuronové sítě fungují, jak se s nimi pracuje a jak se pomocí Pythonu a knihoven TensorFlow upravují data.

#### 5.1.1 Sběr a práce s daty

Dále následovala práce s daty. Pro vytvoření modelu byl preferován kvalitní obrazový dataset obsahující fotografie různých typů kožních infekcí. Obrázky pocházely z odborných portálů o kožních onemocněních a webových stránkách zdravotnických institucí. Bylo důležité, aby obrázky měly vysoké rozlišení a jasně zobrazovaly příznaky konkrétních infekcí.

Dataset byl manuálně upraven. Každý obrázek byl pojmenován tak, aby označoval konkrétní třídu (např. „tinea\_corporis\_1“, „candidiasis\_1“), a následně zařazen do odpovídající složky (třídy). Pro snížení náročnosti na paměť byl celý dataset komprimován do formátu ZIP. Výsledný archiv nesl název „fungi\_dataset“ a obsahoval čtyři hlavní složky: „tinea\_corporis“, „tinea\_pedis“, „tinea\_versicolor“ a „candidiasis“, přičemž každá z nich obsahovala přibližně 40 až 60 fotografií.

Pro účely testování a ukázky byl vytvořen zmenšený dataset „fungi\_dataset\_demo“, který obsahoval 2 až 5 snímků pro každou třídu. Ten sloužil k rychlé prezentaci funkcí modelu nebo k ladění bez nutnosti načítat rozsáhlý soubor.

### 5.2 Nastavení vývojového prostředí

Pro přípravu vývojového prostředí bylo nainstalovány potřebné knihovny a frameworky (TensorFlow a Keras). Práce probíhala v prostředí Jupyter Notebook, konkrétně v jeho online verzi Google Colab.

### 5.3 Práce s ImageDataGenerator

Nahrála jsem dataset, který byl nejprve zabalený ve formátu ZIP, takže bylo třeba jej rozbalit a jednotlivé složky připravit k použití. Následovalo odstranění nevyhovujících či nečitelných fotografií, aby nedocházelo ke zkreslení výsledků.

Velikost fotografií byla sjednocena na  $224 \times 224$  pixelů a byla provedena normalizace. Při normalizaci fotek se barevné obrázky převádějí na jednorozměrný vektor čísel v rozsahu 0–255, kde 0 představuje černou a 255 bílou. Tento rozsah je poměrně široký, což může vést k nestabilitě a složitějším výpočtům. Převodem těchto hodnot na škálu 0–1 se čísla zmenší, čímž se jednorozměrný vektor (číselná řada) stabilizuje. Počítač tak nemusí pracovat s příliš vysokými čísly, data jsou jednotná (na stejné škále) a díky tomu se model učí rychleji a předpovědi jsou přesnější.

Aby se zvýšila variabilita trénovacích dat bez nutnosti shánět nové obrázky, byla použita takzvaná augmentace. Tento proces náhodně mění vlastnosti fotografií, například je otáčí, mění jas, přibližuje či oddaluje. Tím se diverzifikují vstupy, což pomáhá modelu naučit se rozeznávat kožní onemocnění přesněji i při menším počtu originálních fotek.

## 5.4 Stavba CNN

Model postavený na konvoluční neuronové síti (CNN) se neučí obrázky z paměti, ale rozpoznává důležité vizuální znaky, které získává ze zkušeností během tréninku. CNN obsahuje speciální vrstvy; konvoluční, pooling, flatten a dense, které spolupracují na tom, aby ze vstupních obrázků vytáhly klíčové rysy a vzory. Díky této schopnosti je CNN ideální pro klasifikace obrázků.

Konvoluční vrstva přijímá obraz jako vstup, aplikuje na něj filtry a výsledkem jsou tzv. feature mapy, nové obrazy, kde jsou zvýrazněny důležité rysy. Síť se tak učí rozpoznávat například okraje, tvary nebo textury, aniž by bylo nutné tyto rysy ručně označovat.

Pooling vrstva pak tyto výstupy zmenší, zachová přitom nejdůležitější informace a sníží tak množství dat, se kterými síť pracuje. To pomáhá zjednodušit výpočty a zlepšit schopnost modelu generalizovat i na nové obrázky.

Flatten vrstva převede vícerozměrný výstup (například 2D mapy) na jednoduchý jednorozměrný vektor, aby bylo možné tento vektor předat do další vrstvy, která rozhoduje.

Dense vrstva je plně propojená a slouží k samotné klasifikaci. Na základě informací, které byly extrahovány z předchozích vrstev, rozhoduje, do které kategorie (např. tinea corporis nebo candidiasis) obrázek patří. Obvykle tvoří poslední krok modelu.

## 5.5 Trénování

Během trénování model prošel 10 epochami. Epocha označuje jedno úplné projetí trénovacím datasetem. V každé epoše se data zpracovávala po částech (tzv. batchích), což umožňuje efektivnější výpočty i při větších datasetech. Po každé epoše následovala validace modelu, aby

bylo možné sledovat, jak dobře model generalizuje mimo trénovací množinu. Během trénování jsem pozorovala i rychlost exekuce, tedy jak rychle se model učí a predikuje.

```
Epoch 1/10
10/10 ----- 8s 655ms/step - accuracy: 0.3092 - loss: 2.2511 - val_accuracy: 0.4474 - val_loss: 1.3276
Epoch 2/10
10/10 ----- 5s 496ms/step - accuracy: 0.4338 - loss: 1.3426 - val_accuracy: 0.4211 - val_loss: 1.2411
Epoch 3/10
10/10 ----- 5s 456ms/step - accuracy: 0.4013 - loss: 1.2306 - val_accuracy: 0.6053 - val_loss: 1.1450
Epoch 4/10
10/10 ----- 7s 700ms/step - accuracy: 0.6614 - loss: 1.0163 - val_accuracy: 0.5526 - val_loss: 1.1665
Epoch 5/10
10/10 ----- 4s 419ms/step - accuracy: 0.6283 - loss: 0.9029 - val_accuracy: 0.5263 - val_loss: 1.0778
Epoch 6/10
10/10 ----- 7s 588ms/step - accuracy: 0.7479 - loss: 0.6814 - val_accuracy: 0.5526 - val_loss: 1.0612
Epoch 7/10
10/10 ----- 9s 447ms/step - accuracy: 0.8207 - loss: 0.5098 - val_accuracy: 0.5526 - val_loss: 1.1603
Epoch 8/10
10/10 ----- 6s 637ms/step - accuracy: 0.8437 - loss: 0.4304 - val_accuracy: 0.6579 - val_loss: 1.0844
Epoch 9/10
10/10 ----- 5s 462ms/step - accuracy: 0.8871 - loss: 0.2849 - val_accuracy: 0.5789 - val_loss: 1.1604
Epoch 10/10
10/10 ----- 4s 437ms/step - accuracy: 0.9348 - loss: 0.2495 - val_accuracy: 0.6316 - val_loss: 1.5812
```

Obrázek: Print screen výstupu po spuštění trénování modelu. Během jednotlivých epoch lze sledovat, jak se zvyšuje přesnost (accuracy) a zároveň snižuje ztráta (loss), a to jak u trénovacích, tak validačních dat.

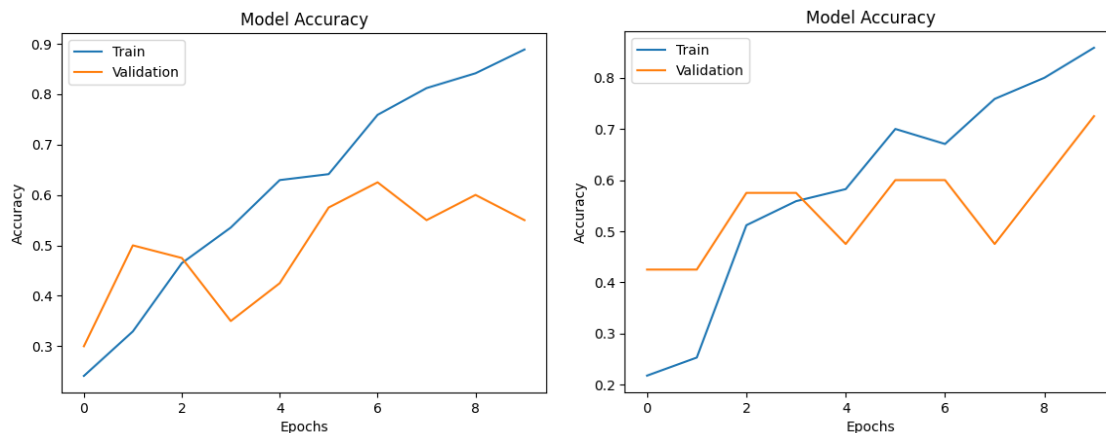
## 5.6 Vyhodnocení modelu a úprava

Při vyhodnocování modelu se objevilo několik problémů, které vyžadovaly opravu a úpravu, například jeden z obrázků byl nesprávně pojmenován jako „fungi\_dataset“, což není kožní infekce, a takové chyby mohly negativně ovlivnit trénink modelu. Kromě toho se v kódu objevily typografické chyby, které bylo nutné odstranit pro správnou funkčnost. Pro zlepšení robustnosti modelu jsem navíc provedla manuální augmentaci dat, zahrnující rotace, změny jasu a ořezávání obrázků, aby model lépe generalizoval a zvládal různé varianty vstupních dat. Tyto zásahy vedly ke zvýšení přesnosti a spolehlivosti modelu.

## 6. Výsledky

Model je obecně funkční, zejména na čistých a kvalitních snímcích, které odpovídají trénovacím datům. Jeho spolehlivost klesá u méně zastoupených tříd a při horší kvalitě obrázků (např. špatné osvětlení, rozmazání nebo atypické úhly). Ukázalo se také, že je citlivý na nekonzistence v datech, jako jsou chybné názvy souborů nebo špatné zařazení do kategorií, což může negativně ovlivnit jeho výstupy. Model zatím zvládá základní rozlišení infekcí, ale pro přesnější a robustnější nasazení by bylo potřeba rozšířit a lépe vyvážit dataset, zlepšit předzpracování a případně implementovat mechanismy pro kontrolu kvality vstupních dat.

Výsledky jsou znázorněny v grafu, kde se křivky přesnosti (accuracy) a u trénovací a validační množiny postupně sblížují po epoše. Z toho vyplývá, že model se každou epochou zlepšuje a úspěšně se učí rozpoznávat obrazové vzory.

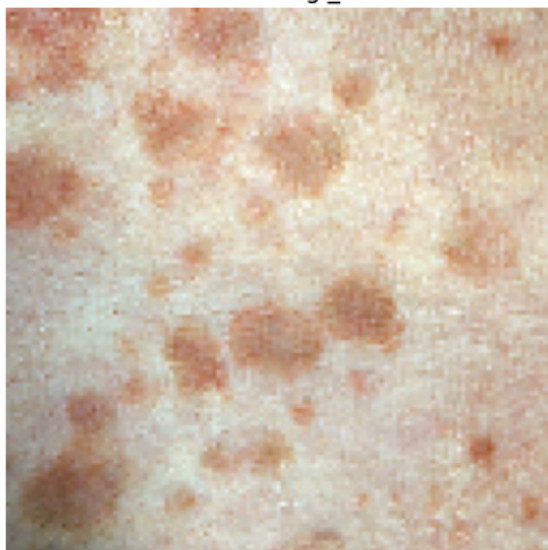


Graf: Dva grafy přesnosti (accuracy) modelu během trénování v průběhu 10 epoch zobrazující vývoj přesnosti na trénovacích datech (modrá čára) a na validačních datech (oranžová čára). Graf vpravo je po druhém trénování. Generováno přímo v Google Colab.



Obrázek: Výstup predikce (správná) na fotografii kožní infekce tinea corporis. Generováno přímo v Google Colab.

Prediction: fungi\_dataset



Obrázek: Výstup predikce (chybná) na fotografii kožní infekce candidiasis. Generováno přímo v Google Colab.

## 7. Závěr

Závěrem lze říci, že se podařilo úspěšně vytvořit a natrénovat model konvoluční neuronové sítě, který dokáže rozpoznat různé typy kožních plísňových infekcí na základě obrázků. Během práce jsem se naučila základy strojového učení, práci s daty, trénink modelu a řešení chyb, které se mohou během vývoje vyskytnout. Pokud bych projekt zopakovala, věnovala bych více času přípravě kvalitního datasetu a lepšímu zpracování vstupních dat. Do budoucna by bylo možné model rozšířit o další třídy onemocnění, zlepšit jeho přesnost pomocí složitější architektury, a nasadit ho do jednoduché webové aplikace, kterou by mohli využít lékaři i pacienti při předběžné diagnostice kožních problémů.

## 8. Literatura

Srov. Solen.cz, 2022, *Tinea corporis u dvouleté dívky*,  
<https://www.solen.cz/pdfs/der/2022/02/10.pdf>

Srov. Mayo Clinic, 2024, *Tinea versicolor*,  
<https://www.mayoclinic.org/diseases-conditions/tinea-versicolor/symptoms-causes/syc-20378385>

Srov. Mayo Clinic, 2023, *Athlete's foot (tinea pedis)*,  
<https://www.mayoclinic.org/diseases-conditions/athletes-foot/symptoms-causes>

Srov. CDC, 2024, *Candidiasis*,  
<https://www.cdc.gov/fungal/diseases/candidiasis/index.html>

Srov. SuperAnnotate, 2023, *What is image classification? Basics you need to know*,  
<https://www.superannotate.com/blog/image-classification-basics>

Srov. Python Software Foundation, 2024, *What is Python? Executive Summary*,  
<https://www.python.org/doc/essays/blurb/>

Srov. Jupyter.org, 2024, *Jupyter Notebook*,  
<https://jupyter.org/>

Srov. TensorFlow, 2024, *Introduction to TensorFlow*,  
<https://www.tensorflow.org/learn>

Srov. Keras, 2024, *Keras: Deep Learning for humans*,  
<https://keras.io/>

Srov. TensorFlow, 2024, *tf.keras.preprocessing.image.ImageDataGenerator*,  
[https://www.tensorflow.org/api\\_docs/python/tf/keras/preprocessing/image/ImageDataGenerator](https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator)

Srov. IBM, 2024, *What are Convolutional Neural Networks?*,  
<https://www.ibm.com/think/topics/convolutional-neural-networksjupyter-notebook.readthedocs.io+3>