

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

AIS ID: 92320

HLADANIE OBJEKTOV NA OBRAZE:
ROZPOZNÁVANIE TVÁRE
ZÁPOČTOVÉ ZADANIE

Predmet:	I-BIOM – Biometria
Prednášajúci:	prof. Dr. Ing. Miloš Oravec
Cvičiaci:	Ing. Marián Šebeňa

Bratislava 2025

Bc. Lukáš Patrnčíak

Obsah

Úvod	1
1 Spracovanie dát	2
2 Metódy na verifikáciu	7
2.1 ResNet50	7
2.2 ArcFace	8
2.3 Analýza a vyhodnotenie	9
3 Architektúra trojičiek	17
4 Vyhodnotenie a porovnanie	18
Záver	21
Zoznam použitej literatúry	22

Zoznam obrázkov a tabuliek

Obrázok 1	Reprezentatívny pár obrázkov rovnakej osoby	3
Obrázok 2	Reprezentatívny pár obrázkov rozdielnej osoby	3
Obrázok 3	Zhody na rovnakých obrázkoch	5
Obrázok 4	Zhody na rozdielnych obrázkoch rovnakej osoby	5
Obrázok 5	Zhody na rôznych obrázkoch rôznej osoby	6
Obrázok 6	ROC krivka ResNet50 a ArcFace modelu	10
Obrázok 7	TRUE najlepšie páry s minimálnou kosinusovou vzdialenosťou 1/3	11
Obrázok 8	TRUE najlepšie páry s minimálnou kosinusovou vzdialenosťou 2/3	11
Obrázok 9	TRUE najlepšie páry s minimálnou kosinusovou vzdialenosťou 3/3	12
Obrázok 10	TRUE najhoršie páry s maximálnou kosinusovou vzdialenosťou 1/3	12
Obrázok 11	TRUE najhoršie páry s maximálnou kosinusovou vzdialenosťou 2/3	13
Obrázok 12	TRUE najhoršie páry s maximálnou kosinusovou vzdialenosťou 3/3	13
Obrázok 13	FALSE napodobnejšie páry s minimálnou kosinusovou vzdialenosťou 1/3	14
Obrázok 14	FALSE napodobnejšie páry s minimálnou kosinusovou vzdialenosťou 2/3	14
Obrázok 15	FALSE napodobnejšie páry s minimálnou kosinusovou vzdialenosťou 3/3	15
Obrázok 16	FALSE najhoršie páry s maximálnou kosinusovou vzdialenosťou 1/3	15
Obrázok 17	FALSE najhoršie páry s maximálnou kosinusovou vzdialenosťou 2/3	16
Obrázok 18	FALSE najhoršie páry s maximálnou kosinusovou vzdialenosťou 3/3	16
Obrázok 19	Konfúzna matica výsledkov z ResNet50	18
Obrázok 20	Konfúzna matica výsledkov z AcrFace	19
Obrázok 21	Konfúzna matica výsledkov z architektúry trojičiek	19
Obrázok 22	ROC krivka výsledkov zo všetkých architektúr	20
Tabuľka 2	Tabuľka výsledkov modelov ResNet50 a ArcFace	10
Tabuľka 3	Porovnanie výsledkov všetkých modelov	18

Zoznam skratiek

AUC	Area Under Curve
CSV	Comma-separated values
FN	False Negatives
FP	False Positives
FPR	False Positive Rate
ReLU	Rectified Linear Unit
ResNet50	Residual Network, 50 vrstiev
ROC	Receiver Operating Characteristic
SIFT	Scale Invariant Feature Transform
TN	True Negatives
TP	True Positives
TPR	True Positive Rate

Úvod

Cielom tejto úlohy je implementovať systém na verifikáciu ľudských tvárí vo vybranom programovacom jazyku (Python). Na riešenie sa využíva dataset FaceVerification, ktorý obsahuje dvojice obrázkov s informáciou o tom, či patria tej istej osobe. Projekt je rozdelený do dvoch častí:

1. **Časť – klasická verifikácia tvárí:** Dataset je načítaný spolu s príslušným CSV súborom. Vykonáva sa analýza počtu zhodných a rozdielnych párov, vrátane vizualizácie reprezentatívnych prípadov. Pomocou algoritmu SIFT sa porovnávajú tri vybrané vzorky tvárí (rovnaké a rozdielne osoby) na úrovni kľúčových bodov. Implementované sú dve metódy extrakcie príznakov:

- ResNet50 trénovaný na ImageNet,
- ArcFace (cez DeepFace) trénovaný na ľudských tvárach.

Porovnanie príznakov sa realizuje pomocou odporúčanej metriky (napr. L2 alebo kosínusová vzdialenosť). Výsledky sú vizualizované pomocou ROC krivky, vypočítaná je AUC a určuje sa optimálny threshold. Nakoniec sa vykonáva vizuálna analýza najlepších a najhorších prípadov pre True/False páry.

2. **Časť 2 – verifikácia pomocou trojičiek (triplet loss):** Na základe výsledkov z časti 1 sa vyberie najlepšie performujúca sieť (ArcFace). Z datasetu sa vytvorí štruktúra trojičiek: Anchor–Positive a Anchor–Negative. Porovnanie sa realizuje znova pomocou ArcFace a výsledky sú vyhodnotené pomocou ROC krivky a konfúznej matice. Upravený dataset vo forme trojičiek je uložený do CSV pre ďalšie využitie.

Tento projekt prepája klasické aj moderné prístupy k rozpoznávaniu tvárí a umožňuje prakticky porovnať výkonnosť rôznych modelov v úlohe verifikácie.

1 Spracovanie dát

Spracovanie dát a implementácia jednotlivých častí zadania bude prebiehať v programovacom jazyku Python¹ s použitím nasledovným knižníc: [1]

1. **OpenCV²**: najpoužívanejšia knižnica na spracovanie obrazu a počítačové videnie
2. **OS³**: práca so súborovým systémom
3. **Matplotlib⁴**: vytváranie vizualizácií a grafov
4. **OS⁵**: štandardná knižnica Pythona na prácu so súbormi a adresármi
5. **NumPy⁶**: knižnica na rýchlu prácu s veľkými poľami (polia, matice) a numerické výpočty.
6. **DeepFace⁷**: knižnica na overovanie tvárí (face verification), rozpoznávanie (face recognition) a analýzu (vek, emócie atď.).
7. **skLearn.metrics⁸**: použitá na výpočet metriky výkonnosti modelov
8. **Pandas⁹**: použitá na čítanie, ukladanie a manipuláciu s tabuľkovými dátami (napr. .csv súbory)
9. **Keras¹⁰**: načítanie a použitie predtrénovaného modelu ResNet50 (trénovaného na ImageNet) ako extraktora príznakov (features)

Najprv bol načítaný dataset a jeho sprievodný CSV súbor¹¹, ktorý obsahuje dvojice obrázkov spolu s informáciou o tom, či patria tej istej osobe. Súbor obsahuje:

- **Zhodných párov**: 3318
- **Rozdielnych párov**: 3339

¹<https://python.org>

²<https://opencv.org/>

³<https://docs.python.org/3/library/os.html>

⁴<https://matplotlib.org/>

⁵<https://docs.python.org/3/library/os.html>

⁶<https://numpy.org/>

⁷<https://pypi.org/project/deepface/>

⁸<https://scikit-learn.org/>

⁹<https://pandas.pydata.org/>

¹⁰<https://keras.io/>

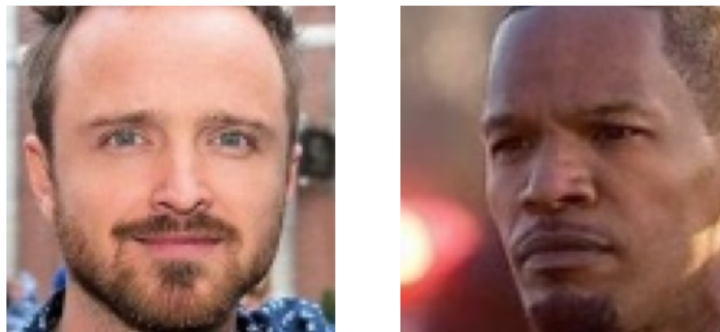
¹¹jednoduchý súborový formát, ktorý sa skladá z riadkov, v ktorých sú jednotlivé položky oddelené znakom. Takýto formát slúži na výmenu tabuľkových dát.

Zhodný pár (rovnaká osoba, iná fotka)
Obrázok 1 Obrázok 2



Obrázok 1: Reprezentatívny pár obrázkov rovnakej osoby

Rozdielny pár (rôzne osoby)
Obrázok 1 Obrázok 2



Obrázok 2: Reprezentatívny pár obrázkov rozdielnej osoby

Ďalej boli hľadané kľúčové body tváre pomocou SIFT algoritmu. Ide o algoritmus, ktorý je robustný voči zmenám mierky, rotácie, jasú a čiastočne aj voči deformáciám a šumu. Používa sa najmä v úlohách počítačového videnia ako sú:

- zladenie obrazov (image matching),
- rozpoznávanie objektov,
- odhad pohybu alebo štruktúry.

Stručný popis činnosti algoritmu:

- *Detekcia kľúčových bodov* – identifikuje miesta s výraznou štruktúrou (napr. hrany, rohy).
- *Popis bodov (deskriptory)* – pre každý bod vypočíta vektor, ktorý reprezentuje lokálne okolie.

- *Porovnanie bodov* – pomocou metriky (L2 vzdialenosť) sa vyhľadávajú zhody medzi dvoma obrázkami.

L2 vzdialenosť (alebo Euklidovská vzdialenosť) je štandardná metrika na meranie „vzdialenosti“ medzi dvoma bodmi vo viacerých rozmeroch. Používa sa na porovnávanie vektorov príznakov, ktoré generuje model z obrázku tváre.

Majme body $A = [a_1, a_2, \dots, a_n]$ a $B = [b_1, b_2, \dots, b_n]$. Potom L2 vzdialenosť týchto bodov je

$$L2(A, B) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots + (a_n - b_n)^2}.$$

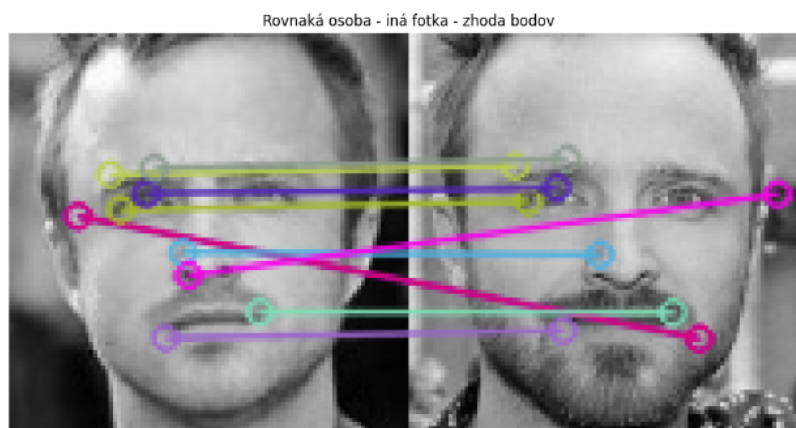
Postup pri vykonaní analýzy zhodnosti bodov medzi obrázkami tváří pomocou SIFT algoritmu:

1. Vybrať páry z datasetu: prvý zhodný pár (target = 1) a prvý rozdielny pár (target = 0) zo súboru dataframe.
 2. Načítať obrázky daných párov a skonverovať ich do odtieňov sivej (grayscale), pričom výstup je 2D pole.
 3. Vytvoriť SIFT objekt: SIFT extraktor (zoznam keypointov + deskripty).
 4. Zhodnotenie zhody na obrázku: Detekcia kľúčových bodov a výpočet deskriptorov pomocou SIFT algoritmu. Najprv deteguje zaujímavé body v obrázku (napr. rohy, hrany), potom pre každý bod vypočíta vektor vlastností (deskriptor). Výstupom sú keypoints: zoznam objektov KeyPoint a descriptors: matica s vektormi (napr. 128-rozmerné pre SIFT) Detekuje kľúčové body (keypoints) a vypočíta ich deskripty pomocou SIFT algoritmu.
 5. Nájdenie potenciálnych zhôd medzi dvoma obrázkami pomocou deskriptorov. Používa sa Brute Force Matcher, pričom KNN Match nájde dva najbližšie zhody pre každý deskriptor. Následne sa aplikuje Lowe's ratio test: ak je vzdialenosť prvej zhody výrazne menšia než druhej ($m.distance < TRESHOLD \cdot n.distance$), považuje ju za „dobrú zhodu“. Výstupom je zoznam „dobrých“ zhôd.
- Rovnaký obrázok: 72 zhôd (dentický obrázok obsahuje prakticky rovnaké kľúčové body, preto je počet zhôd veľmi vysoký)



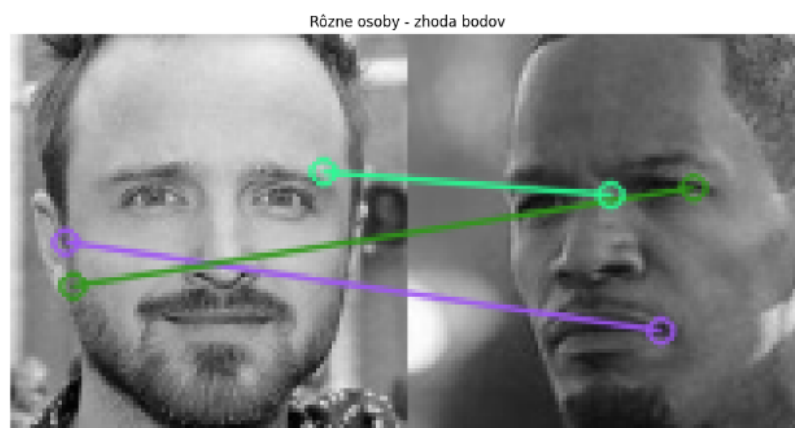
Obrázok 3: Zhody na rovnakých obrázkoch

- Rovnaká osoba na obrázku, iná fotka: 10 zhôd (medzi dvoma rôznymi fotkami tej istej osoby bývajú zmeny v uhle, výraze alebo osvetlení, preto počet zhôd klesá)



Obrázok 4: Zhody na rozdielnych obrázkoch rovnakej osoby

- Rôzne osoby na obrázkoch: 3 zhody (nízky počet zhôd je žiadaný, pretože osoby sú odlišné – ukazuje to, že SIFT ich vie do určitej miery rozlíšiť)



Obrázok 5: Zhody na rôznych obrázkoch rôznej osoby

2 Metódy na verifikáciu

Pre verifikáciu tvári boli vybrané 2 ľubovoľné metódy, pričom jedna je tréňovaná na ImageNete a druhá je tréňovaná na ľudských tvárach.

2.1 ResNet50

Residual Network, 50 vrstiev (ResNet50) je hlboká konvolučná neurónová sieť. Jej hlavná inovácia spočíva v tzv. "residual connections" – teda prepojeniach, ktoré obchádzajú niektoré vrstvy. Tieto prepojenia umožňujú ľahší prenos gradientov počas spätného šírenia a umožňujú tréňovať veľmi hlboké siete bez degradácie výkonu. ResNet50 pozostáva zo základných stavebných blokov zložených z:

- konvolučných vrstiev,
- batch normalizácie,
- ReLU aktivácií,
- a priamych prechodov (skip connections).

Rectified Linear Unit (ReLU) je aktivačná funkcia definovaná ako: $f(x) = \max(0, x)$. Zabezpečuje nelinearitu v neurónových sieťach a zároveň je výpočtovo efektívna a pomáha zabráňovať saturácii gradientov, čím uľahčuje učenie.

Popis krokov pri extrakcii hlbokého príznakového vektora:

1. Načíta obrázok
2. Zmení jeho veľkosť na požadovaný rozmer
3. Konvertuje z BGR (OpenCV štandard) na RGB (štandard pre Keras)
4. Pridá extra dimenziu (batch size = 1), aby sa hodil ako vstup do modelu
5. Normalizuje obraz
6. Preženie obraz modelom pomocou predikcie
7. Výstup (embedding) sploštuje do 1D vektora

Výsledkom je numerický vektor reprezentujúci obraz, ktorý môžeš porovnávať s inými (cez L2 vzdialenosť).

- **Tréningová databáza:** Použitá verzia modelu bola predtréňovaná na *ImageNet* – veľkom datasete obsahujúcom vyše 1 milión obrázkov z 1000 kategórií objektov.
- **Odporúčaná metrika:** Porovnávanie embeddingov pomocou *L2 vzdialenosti*.

2.2 ArcFace

ArcFace je model pre rozpoznávanie tvárí, ktorý sa preslávil vďaka svojej presnosti a robustnosti. Vznikol ako riešenie problému, ako lepšie odlíšiť osoby na základe ich tvárí, aj keď sa ich výzor mení (výraz, uhol, svetlo). Namiesto obvyčajnej klasifikačnej straty používa tzv. *Additive Angular Margin Loss*, ktorá zväčšuje uhlové vzdialenosti medzi triedami (osobami).

- Embeddingy z ArcFace sú *L2-normalizované* a ležia na jednotkovej hyperguli.
- Táto vlastnosť umožňuje presné a spoľahlivé porovnávanie tvárí pomocou uhlových metód.

Embedding je číselný vektor (napr. 512-dimenzionálny), ktorý reprezentuje tvár v číselnom priestore tak, aby podobné tváre boli blízko seba a rozdielne tváre boli ďalej od seba.

L2-normalizácia znamená, že každý embedding je upravený tak, aby jeho dĺžka bola rovná 1. Matematicky:

$$||v||_2 = \sqrt{v_1^2 + v_2^2 + \dots + v_n^2} = 1.$$

Tým pádom všetky embeddingy ležia na povrchu jednotkovej sféry v n-rozmernom priestore (tzv. hypergula - generalizácia 3D gule do viacdimenzionálneho priestoru, pričom všetky vektory ležia na povrchu takejto gule s polomerom 1) a môžu sa porovnávať len podľa smeru, nie podľa veľkosti.

Typická ArcFace implementácia využíva backbone sieť¹² *ResNet50*. Výstupom je 512-dimenzionálny embedding pre každú tvár.

Zhrnutie krokov ArcFace:

1. Vstup = obrázok tváre
 2. CNN (napr. ResNet50) vytvorí 512-dimenzionálny embedding (vektor)
 3. Na tento vektor sa aplikuje ArcFace loss, ktorá:
 - udržiava tváre tej istej osoby blízko pri sebe (malý uhol)
 - a tváre rôznych osôb ďaleko od seba (väčší uhol + margin)
- **Tréningová databáza:** V prípade použitia *DeepFace* (v tomto prípade) alebo *InsightFace* sa najčastejšie používa predtrénovaný model *buffalo_l* alebo *adaface* trénovaný na *Glint360K*.

¹²predtrénovaná (často konvolučná) neurónová sieť, ktorá extrahuje črty zo vstupu. Následne tieto črty môžu byť použité na klasifikáciu, detekciu, verifikáciu a ďalšie úlohy.

- **Odporúčaná metrika:** *Kosinusová vzdialenosť (podobnosť)*

Kosinusová vzdialenosť meria uhlovú medzi dvoma vektormi v priestore bez ohľadu na ich veľkosť. Matematicky:

$$\text{cosine_similarity}(A, B) = \frac{A \cdot B}{\|A\| \cdot \|B\|},$$

$$\text{cosine_distance} = 1 - \text{cosine_similarity}$$

Výsledok:

- 1 - úplne odlišné vektory (uhol 180°),
- 0 - identické vektory (uhol 0°),
- Hodnota medzi 0 a 1 - miera rozdielnosti.

Porovnanie s L2 vzdialenosťou:

- L2 vzdialenosť - meria priamu vzdialenosť medzi dvoma bodmi vo viacerých dimenziách.
- Kosínusová podobnosť - zameriava sa na smer vektorov (uhlovo), nie ich veľkosť.

2.3 Analýza a vyhodnotenie

V tejto podkapitole bude prebiehať analýza a vyhodnotenie výsledkov na základe ROC krivky. Receiver Operating Characteristic (ROC) krivka je graf, ktorý znázorňuje vzťah medzi:

- **True Positive Rate (TPR)** – citlivosť

$$TPR = \frac{TP}{TP + FN} = \frac{\text{počet skutočne správne určených pozitívnych prípadov}}{\text{súčet všetkých skutočných pozitívnych prípadov}}$$

- **False Positive Rate (FPR)** – počet falošne pozitívnych oproti všetkým negatívnym

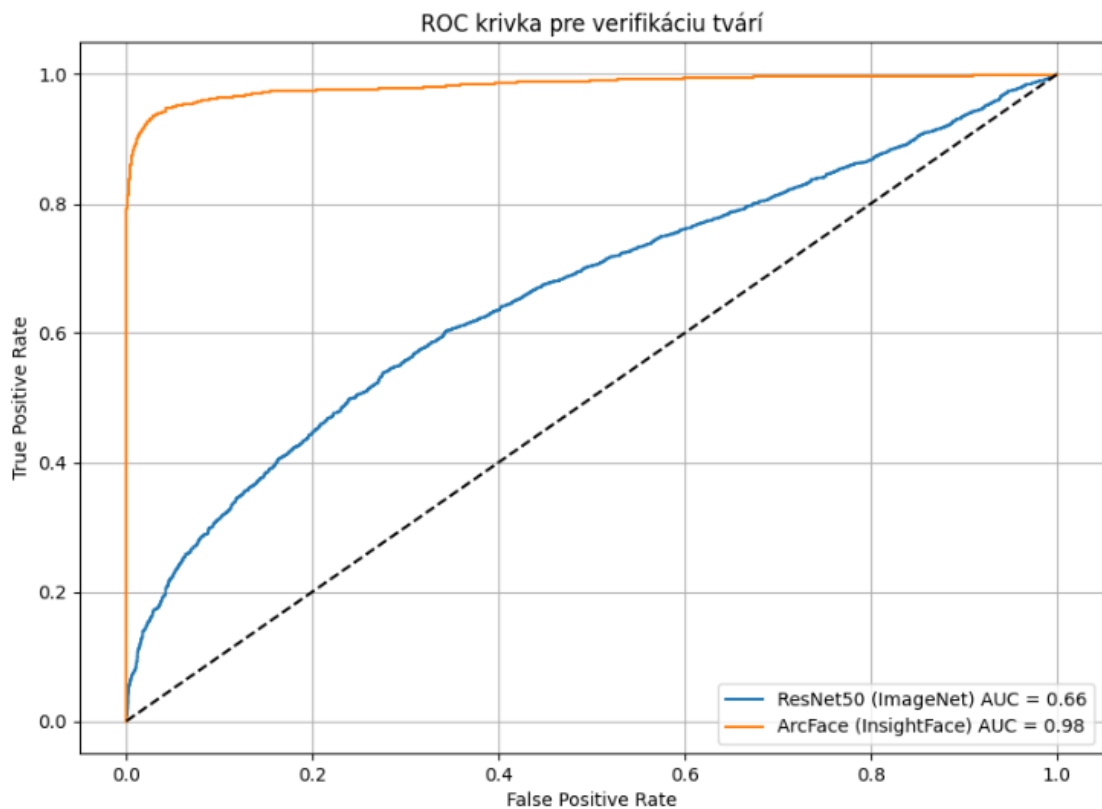
$$FPR = \frac{FP}{FP + TN} = \frac{\text{počet nesprávne označených prípadov ako pozitívny}}{\text{súčet všetkých skutočných negatívnych prípadov}}$$

Používa sa na vizuálne vyhodnotenie výkonnosti binárneho klasifikátora pri rôznych hodnotách thresholdu. Oblasť pod krivkou *Area Under Curve (AUC)* predstavuje číselné vyjadrenie kvality klasifikátora. Čím viac sa krivka blíži k ľavému hornému rohu, tým lepšia je výkonnosť modelu. **Popis ďalších výpočtov:**

- *Najlepší spôsob generovania príznakov:* V tomto prípade budeme vyberať najlepší model na základe AUC (čím vyššie, tým lepšie) (obr. 6 a tab. 2)

- *Optimálny treshold*: Vypočíta sa ako $TPR - FPR$. Určuje, či sú tváre "rovnaké" (zhodujú sa), a kedy sú "rôzne".
 - skóre \geq threshold \rightarrow predikcia = zhodné(1)
 - skóre $<$ threshold \rightarrow predikcia = nezhodné(0)
- *Thresholds*: týka sa prahových hodnôt, ktoré sa používajú na rozdelenie skóre na triedu 0 alebo 1, aby sa vypočítali hodnoty TPR a FPR pre ROC krivku.

Špeciálny prípad ResNet50 modelu: v tomto prípade používa L2 vzdialenosť, pričom platí, že čím nižšia L2 vzdialenosť, tým vyššia pobovnosť. Avšak `roc_curve()` predpokladá, že čím väčšia hodnota skóre, tým väčšia pravdepodobnosť triedy 1 (zhoda) a preto je potrebné dať zápornú L2 vzdialenosť.



Obrázok 6: ROC krivka ResNet50 a ArcFace modelu

Model	AUC	Optimálny treshold
ResNet50	0.6839	13.1189
AcrFace	0.9835	0.2429

Tabuľka 2: Tabuľka výsledkov modelov ResNet50 a ArcFace

Na základe dosiahnutých výsledkov je možné konštatovať, že najlepšie obstál ArcFace model, čiže **ArcFace je vhodnejší spôsob generovania príznakov**. Pomocou tohto modelu a jeho metriky budeme teda analyzovať 3 najlepšie a najhoršie páry pre rovnaké osoby (TRUE) a rozdielne osoby (FALSE):

1. **TRUE páry osôb:**

- Najlepšie páry (minimálna kosinová vzdialenosť): Tieto obrázky majú spoločné to, že sa na nich nachádza rovnaká osoba, pričom aj farby na obrázku sú približne rovnaké.

TRUE (najlepšie - min. vzdialenosť): Vzdialenosť 0.9792
Obrázok 1 Obrázok 2



Obrázok 7: TRUE najlepšie páry s minimálnou kosinovou vzdialenosťou 1/3

TRUE (najlepšie - min. vzdialenosť): Vzdialenosť 0.9718
Obrázok 1 Obrázok 2



Obrázok 8: TRUE najlepšie páry s minimálnou kosinovou vzdialenosťou 2/3

TRUE (najlepšie - min. vzdialenosť): Vzdialenosť 0.9715
Obrázok 1 Obrázok 2



Obrázok 9: TRUE najlepšie páry s minimálnou kosinusovou vzdialenosťou 3/3

- Najhoršie páry (maximálna kosinusová vzdialenosť): Ide o obrázok tých istých osôb, avšak s rozdielnym výzorom, pričom mierne zhoršenie nastalo aj vo farbách na obrázkoch.

TRUE (najhoršie - max. vzdialenosť): Vzdialenosť -0.1214
Obrázok 1 Obrázok 2



Obrázok 10: TRUE najhoršie páry s maximálnou kosinusovou vzdialenosťou 1/3

TRUE (najhoršie - max. vzdialenosť): Vzdialenosť -0.1244
Obrázok 1 Obrázok 2



Obrázok 11: TRUE najhoršie páry s maximálnou kosinusovou vzdialenosťou 2/3

TRUE (najhoršie - max. vzdialenosť): Vzdialenosť -0.1496
Obrázok 1 Obrázok 2

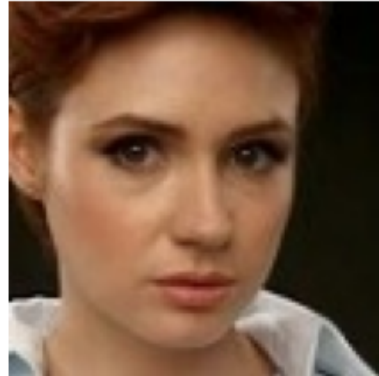
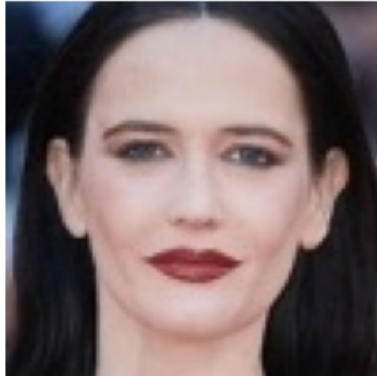


Obrázok 12: TRUE najhoršie páry s maximálnou kosinusovou vzdialenosťou 3/3

2. FALSE páry osôb:

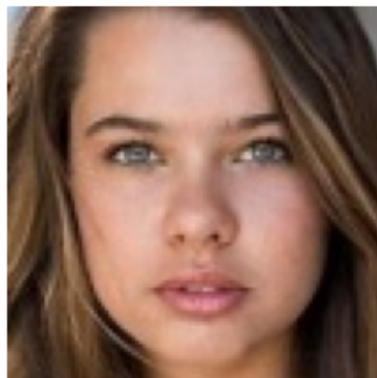
- Najlepšie páry (minimálna kosinusová vzdialenosť): Potenciálna falošná zhoda (podobnosť) osôb - na týchto obrázkoch sa nachádzajú obrázky tváří, ktoré vykazujú isté známky podobných črt.

FALSE (potenciálna falošná zhoda - min. vzdialenosť): 0.4183



Obrázok 13: FALSE napodobnejšie páry s minimálnou kosinusovou vzdialenosťou 1/3

FALSE (potenciálna falošná zhoda - min. vzdialenosť): 0.3885



Obrázok 14: FALSE napodobnejšie páry s minimálnou kosinusovou vzdialenosťou 2/3

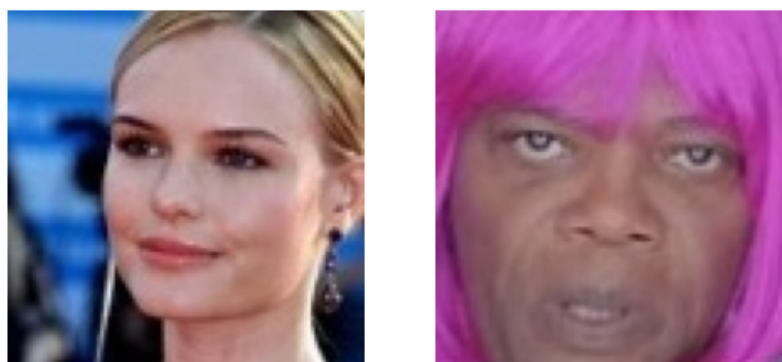
FALSE (potenciálna falošná zhoda - min. vzdialenosť): 0.3816
Obrázok 1 Obrázok 2



Obrázok 15: FALSE napodobnejšie páry s minimálnou kosinusovou vzdialenosťou 3/3

- Najhoršie páry (maximálna kosinusová vzdialenosť): Potenciálne dobré rozlíšenie daných osôb - na týchto obrázkoch sa nachádzajú páry, ktoré nevykazujú žiadne podobnosti.

FALSE (potenciálne dobré rozlíšenie - max. vzdialenosť): -0.2058
Obrázok 1 Obrázok 2



Obrázok 16: FALSE najhoršie páry s maximálnou kosinusovou vzdialenosťou 1/3

FALSE (potenciálne dobré rozlíšenie - max. vzdialenosť): -0.2280
Obrázok 1 Obrázok 2



Obrázok 17: FALSE najhoršie páry s maximálnou kosinusovou vzdialenosťou 2/3

FALSE (potenciálne dobré rozlíšenie - max. vzdialenosť): -0.2655
Obrázok 1 Obrázok 2



Obrázok 18: FALSE najhoršie páry s maximálnou kosinusovou vzdialenosťou 3/3

3 Architektúra trojičiek

Ako je možné z predchádzajúcej analýzy dát z kapitoly 2.3 vidieť, najlepšia performujúca backbone sieť pre verifikáciu tváří bola na základe dosiahnutých výsledkov **ArcFace**, pomocou ktorej bude realizovaná verifikácia tváří architektúrou trojičiek. Trojičková architektúra sa zakladá na učení pomocou trojíc obrázkov:

- **Anchor** – referenčný obrázok osoby, ktorú chceme overiť
- **Positive** – iný obrázok tej istej osoby ako Anchor
- **Negative** – obrázok inej osoby

Cieľom architektúry je naučiť model mapovať obrázky tváří do embeddingového priestoru tak, aby:

- vzdialenosť medzi Anchor a Positive bola *menšia* ako medzi Anchor a Negative,
- a zároveň existovala určitá minimálna *margin* medzi týmito vzdialenosťami.

Vzdialenostná podmienka:

$$\|f(\text{Anchor}) - f(\text{Positive})\|_2^2 + \alpha < \|f(\text{Anchor}) - f(\text{Negative})\|_2^2,$$

kde α je *margin* – bezpečnostný odstup.

Postup vytvorenia nových dvojíc: Pre každý index i :

1. Vezme anchor obrázok (z image_1 zhodného páru)
2. Vytvorí:
 - zhodnú dvojicu (anchor, positive) - trieda 1
 - nezhodnú dvojicu (anchor, negative) - trieda 0
3. Výsledkom vzniknú dve nové dvojice pre každý index:
 - (image_1_A, image_2_A, 1)
 - (image_1_A, image_2_B, 0)

Táto architektúra zvyšuje diskriminačnú schopnosť embeddingov a umožňuje učenie bez nutnosti explicitného označovania všetkých možných tried (osôb), pričom bolo potrebné upraviť pôvodný dataset nasledovným spôsobom (bol taktiež aj exportovaný nový CSV súbor s týmito dátami):

- Z pôvodných dvojíc vznikli nové páry so spoločným „anchor“,
- Vzniká vyvážený dataset (rovnaký počet pozitívnych a negatívnych prípadov),
- Dataset je teraz pripravený na verifikáciu modelu na princípe „je to rovnaká osoba alebo nie?“ — s použitím triplet logiky.

4 Vyhodnotenie a porovnanie

V tejto kapitole bude popísané vyhodnotenie výsledkov doiahnutých architektúrou dvojčiek a taktiež aj porovnanie výsledkov s predchádzajúcimi modelmi (ResNet50 a AcrFace) na základe ROC krivky a konfúznej matice.

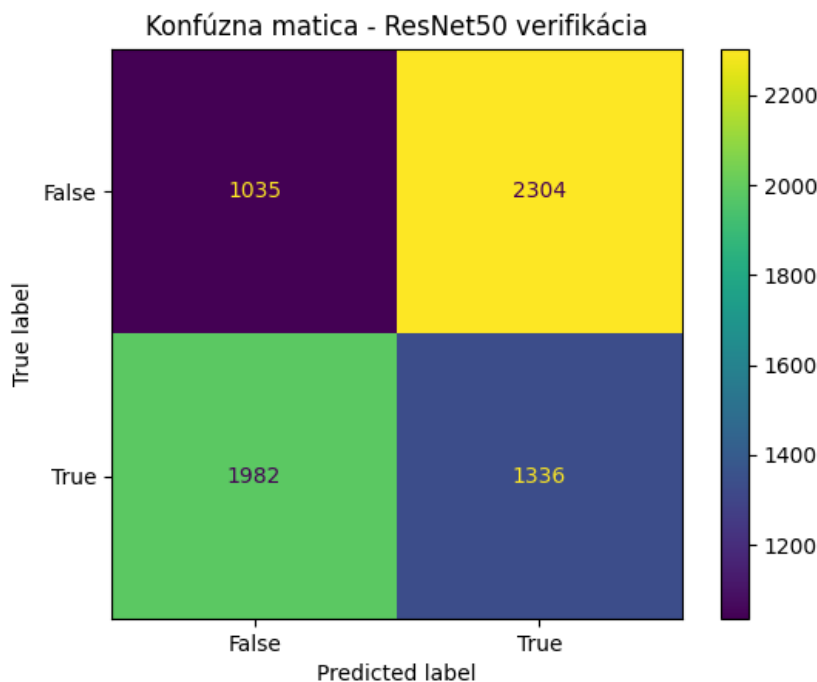
Konfúzna matica je tabuľka, ktorá znázorňuje skutočné a predikované kategórie klasifikátora. Umožňuje analyzovať typy chýb modelu:

- **True Positives (TP)** – správne predikované zhodné páry
- **True Negatives (TN)** – správne predikované rôzne páry
- **False Positives (FP)** – falošné zhody (model tvrdí, že sú rovnakí, ale nie sú)
- **False Negatives (FN)** – falošné rozdiely (model tvrdí, že sú rôzni, ale sú rovnakí)

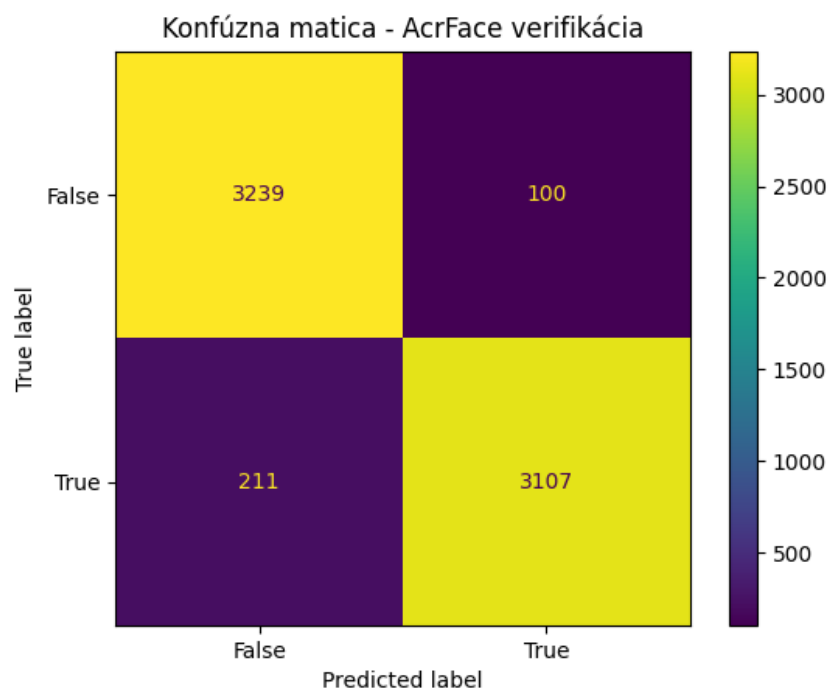
	ResNet50	AcrFace	Trojičková architektúra
AUC	0.6839	0.9835	0.9813
Optim. Threshold	13.1189	0.2429	0.2546

Tabuľka 3: Porovnanie výsledkov všetkých modelov

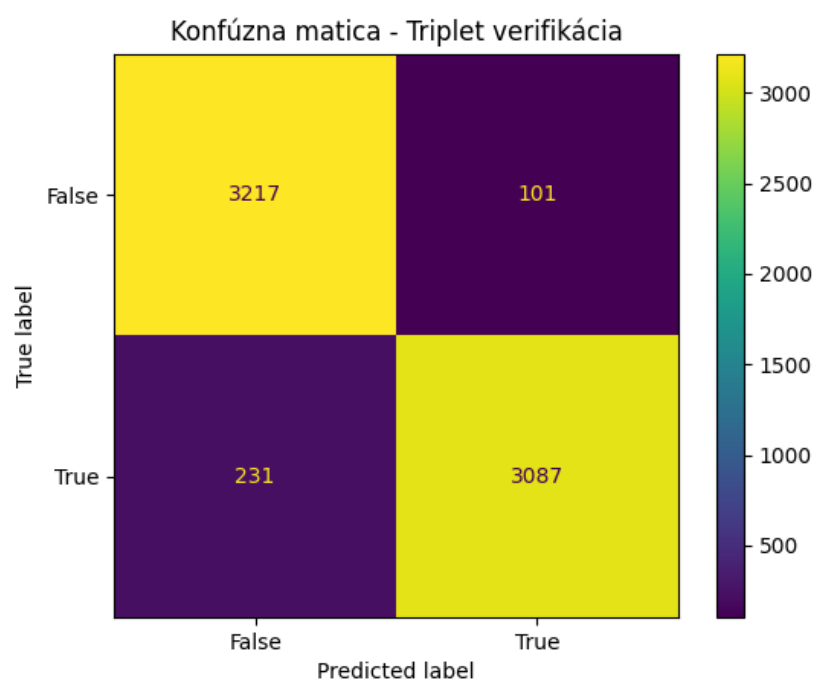
Vyšší threshold (prahová hodnota) znamená, že sme prísnejší pri rozhodovaní, keď povieme, že dve tváre (alebo vstupy) patria tej istej osobe (trieda 1 – zhoda).



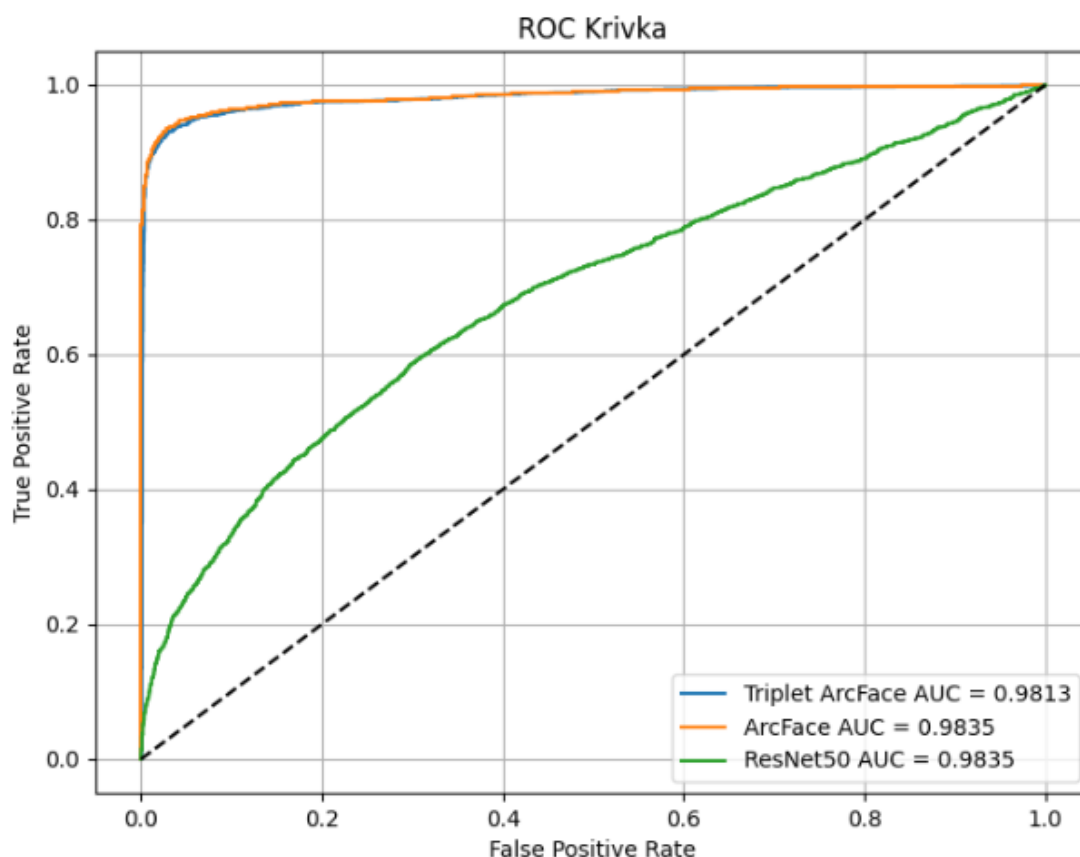
Obrázok 19: Konfúzna matica výsledkov z ResNet50



Obrázok 20: Konfúzna matica výsledkov z AcrFace



Obrázok 21: Konfúzna matica výsledkov z architektúry trojičiek



Obrázok 22: ROC krivka výsledkov zo všetkých architektúr

Na základe ROC krivky a konfúzných matíc (správne predikcie podľa hlavnej diagonály), je možné skonštatovať, že model ArcFace je výrazne presnejší a spoľahlivejší pri verifikácii tváří než klasický ResNet50, ktorý nebol tréňovaný špecificky na tváre.

ArcFace dosahuje vysoké hodnoty AUC (0.98), vrátane použitia trojičkovej architektúry (ktorá je len o malý zlomok hodnoty "horšia"). Naopak, ResNet50 dosahuje len slabú výkonnosť (AUC 0.66), čo naznačuje, že nie je vhodný na túto úlohu bez špecializovaného pretrénovania.

Pre úlohy biometrickej verifikácie jednoznačne odporúčame ArcFace alebo iný model špecificky tréňovaný na tvárové embeddingy.

Záver

Cieľom tohto projektu bolo navrhnúť a implementovať systém na verifikáciu ľudských tvárí pomocou moderných prístupov strojového učenia. Na analýzu a porovnanie boli použité dve metódy: ResNet50 trénovaný na ImageNet a ArcFace trénovaný špeciálne na rozpoznávanie tvárí. Ich výkonnosť bola porovnaná pomocou ROC kriviek, AUC skóre a konfúzných matíc.

Výsledky jasne ukázali, že ArcFace model dosiahol najvyššiu presnosť ($AUC = 0.9835$) a je výrazne vhodnejší na úlohy biometrickej verifikácie v porovnaní s ResNet50 ($AUC = 0.6839$), ktorý nebol trénovaný na tvárové dáta. Okrem toho bola implementovaná aj architektúra trojičiek, ktorá ďalej podporila diskriminačnú schopnosť embeddingov a umožnila efektívnejšie učenie podobností medzi obrazmi.

Projekt prepojil klasické prístupy (SIFT) s modernými hlbokými modelmi, čím poskytol komplexný pohľad na problematiku rozpoznávania tvárí. Na základe vykonanej analýzy je možné odporučiť využitie ArcFace alebo iných modelov trénovaných na tvárové dáta ako optimálne riešenie pre biometrickú verifikáciu v praxi

Zoznam použitej literatúry

1. HALVORSEN, H. P. *Python Programming*. Dostupné tiež z: <https://www.helvorsen.blog/documents/programming/python/resources/Python%20Programming.pdf>.