



Univerzita Komenského v Bratislave  
Fakulta matematiky, fyziky a informatiky



Autoreferát dizertačnej práce

# Invariant Visual Object Recognition Using Hierarchical Temporal Memory

Škálovo a pozične invariantné rozpoznávanie vizuálnych objektov s využitím  
biologicky inšpirovaného modelu HTM siete

Mgr. Kristián Valentín

na získanie akademického titulu *philosophiae doctor*  
v odbore doktorandského štúdia: 9.2.1 Informatika

Bratislava, 2014

Dizertačná práca bola vypracovaná v dennej forme doktorandského štúdia na Katedre aplikovanej informatiky FMFI UK.

**Predkladateľ:** Mgr. Kristián Valentín  
Katedra aplikovanej informatiky  
Fakulta matematiky, fyziky a informatiky  
Univerzita Komenského v Bratislave  
Mlynská dolina, 842 48 Bratislava

**Školiteľ:** prof. RNDr. Ing. Ivan Bajla, PhD.  
Ústav merania  
Slovenská akadémia vied  
Dúbravská cesta 9, 841 04 Bratislava

**Oponenti:** prof. RNDr. Jiří Pospíchal, DrSc.  
Ústav aplikovanej informatiky  
Fakulta informatiky a informačných technológií  
Slovenská technická univerzita v Bratislave

prof. Dr. Ing. Miloš Oravec  
Ústav informatiky a matematiky  
Fakulta elektrotechniky a informatiky  
Slovenská technická univerzita v Bratislave

RNDr. Barbara Zitová, PhD.  
Oddelení zpracování obrazové informace  
Ústav teorie informace a automatizace  
Česká akademie věd

Obhajoba dizertačnej práce sa koná ..... o ..... h pred komisiou pre obhajobu dizertačnej práce v odbore doktorandského štúdia vymenovanou predsedom odborovej komisie študijného programu 9.2.1 Informatika na .....

**Predseda odborovej komisie:**

prof. RNDr. Branislav Rován, PhD.  
Fakulta matematiky, fyziky a informatiky  
Univerzita Komenského v Bratislave

# 1 Úvod

## 1.1 Motivácia

Existuje veľa zaujímavých otvorených otázok čakajúcich na možno ešte zaujímavejšie odpovede. Toto platí nielen pre oblasť umelej inteligencie (UI). V prípade UI je to ale predsa len možno o niečo iné. UI sa snaží skúmať, dosiahnuť i prekonať ľudskú inteligenciu. A práve tento ľudský faktor prináša veľkú pozornosť vedeckej komunity nielen z pohľadu možných aplikácií, ale tiež z pohľadu získania poznatkov o našej vlastnej inteligencii.

Jedna z otvorených otázok v oblasti videnia, ľudského videnia osobitne, je otázka: ako vlastne vidíme? Ako rozpoznávame objekty? A najmä, ako v počítači dosiahnuť úroveň ľudského rozpoznávania objektov? Je to vôbec možné? Dá sa ľudské videnie formalizovať či algoritmizovať? Pri hľadaní odpovedí treba kombinovať výskum vo viacerých vedeckých disciplínach – neurovede, psychológii či informatike.

Jeff Hawkins v tomto kontexte navrhol niekoľko premís:

1. Princípy fungovania mozgu sa dajú pochopiť.
2. Vieme zostrojiť stroje, ktoré budú pracovať na týchto princípoch.
3. Veľa problémov v oblasti strojového učenia, UI a robotiky je možné vyriešiť len týmto spôsobom.

## 1.2 Čo je kortex a prečo ho skúmať?

Kortex je evolučne najmladšia časť mozgu cicavcov, kde sa vykonávajú všetky kognitívne procesy, ako je pamäť, pozornosť, plánovanie, senzorická percepcia, rozhodovanie i rozpoznávanie vizuálnych a iných objektov. Pri hľadaní inšpirácie pre riešenie úlohy rozpoznávania je preto prirodzené skúmať práve kortex.

Kortex obsahuje niekoľko vrstiev neurónov. Kortex cicavcov má 6 vrstiev. Každá z nich má svoju charakteristickú štruktúru a funkciu, čo je špecifické pre rôzne kortikálne oblasti. Napriek tejto rozmanitosti existuje niekoľko všeobecných spoločných princípov a charakteristík, ako napríklad informačný tok v rámci i medzi vrstvami, ktoré sa dajú pozorovať naprieč kortexom.

Kortex je hierarchická štruktúra, a to nielen lokálne medzi vrstvami, ale aj globálne v rámci jednotlivých kortikálnych oblastí. Prvé dôkazy pre tento fakt pochádzajú zo štúdií vizuálneho kortexu makakov [FVE91]. Dve veci sa dejú, keď sa pozeráme na hierarchiu zospodu nahor: *konvergencia informácií* a *časové spomalenie*. Konvergencia informácií znamená, že keď sa vzdalujeme od senzorického vstupu, neuróny reprezentujú čoraz väčšiu časť vstupného priestoru. Časové spomalenie znamená, že neuróny blízko senzorického vstupu spracúvajú rýchlo sa meniaci vstup, zatiaľ čo neuróny vyššie v hierarchii reagujú v oveľa dlhších časových škálach. Napríklad auto prechádzajúce cez zorné pole očí spôsobí rýchle zmeny vo farbe i intenzite svetla na sietnici, na čo musia reagovať

blízke neuróny veľmi rýchlo. Po krátkom čase neuróny vyššie v hierarchii identifikujú časti auta (napríklad kolesá, dvere, či okná). Nakoniec, na vrchole hierarchie príde k výslednej klasifikácii (kategorizácii), že ide o auto. Táto klasifikácia sa však nezmení, aj keď je auto stále v pohybe. Reprezentácie v týchto neurónoch sú v čase stabilnejšie.

Aj napriek rôznej morfológii a funkcii kortikálnych oblastí bola navrhnutá spoločná funkcionálna štruktúra kortexu [Cre77, Mou78]. Hlavná myšlienka spočíva v tom, že v rôznych kortikálnych oblastiach sa dá pozorovať *spoločný algoritmus* alebo rovnaký spôsob spracúvania vstupných informácií. Napríklad kortikálne oblasti zodpovedné za videnie a kortikálne oblasti zodpovedné za pociťovanie vykonávajú v princípe ten istý algoritmus. Jediný zásadný rozdiel medzi nimi je v charaktere ich vstupných informácií. Podporný dôkaz pre túto hypotézu pochádza z experimentov na fredkách, v ktorých boli presmerované neurálne dráhy z vizuálneho kortexu do sluchového kortexu [RPHS90]. Po nejakom čase boli fredky schopné “vidieť” ich sluchovým kortexom. Iný druh experimentu zahŕňal použitie poľa 49 elektród pripojených k jazyku človeka, ten bol schopný po čase v obmedzenom množstve vidieť pomocou kamery prepojenej s elektródami [ByRKT<sup>+</sup>98].

Vlastnosti kortexu, ktoré umožňujú ľuďom (a iným vyšším primátom) riešiť náročné kognitívne úlohy, môžu slúžiť ako inšpirácia pre vytvorenie podobne fungujúcich algoritmov. Preto je aj z pohľadu umelej inteligencie štúdium kortexu dôležitá výskumná úloha.

## 2 Ciele práce

Cielom dizertačnej práce je výskum vlastností modelu HTM pre aplikáciu v úlohe invariabilného rozpoznávania vizuálnych objektov. Rozpoznávanie by malo byť invariantné voči posunutiu, rotácii a zmene priestorovej škály objektov. Zároveň je ambíciou práce zistiť jednotlivé slabé miesta pôvodného modelu HTM v tejto úlohe a navrhnúť riešenia pre ich odstránenie.

Hlavné ciele práce sú:

- vývoj vlastnej implementácie algoritmov modelu HTM a podporných nástrojov pre umožnenie zmien v týchto algoritmoch a pre umožnenie vykonávania počítačových experimentov,
- výskum ako je model HTM schopný zlepšiť invariantné vlastnosti vstupných obrazových dát,
- vývoj nového algoritmu na generovanie tréningových sekvencií pre časové učenie sekvencií zo statických obrazov,
- porovnanie modelu HTM s inými aktuálnymi bioinšpirovanými modelmi určenými pre rozpoznávanie obrázkov, ako napríklad Deep Belief Networks (DBN), Hierarchical model and X (HMAX) alebo Convolutional Neural Networks (CNNs),
- a napokon štúdium spätného šírenia informácií v modeli HTM pre použitie na rozpoznávanie vizuálnych objektov.

### 3 Hierarchická časová pamäť

Hierarchická časová pamäť (angl. Hierarchical Temporal Memory, skratka HTM) je výpočtový model kortexu predstavený v [HB04, GH05, GH09] spoločnosťou Numenta, Inc<sup>1</sup>.

HTM sa snaží zachytiť štrukturálne a algoritmické vlastnosti kortexu s cieľom dosiahnuť v rôznych kognitívnych úlohách (napríklad rozpoznávanie vizuálnych objektov) výsledky porovnateľné s ľudskými.

Na rozdiel od iných biologicky inšpirovaných modelov, HTM kombinuje priestorové a časové učenie. Priestorové učenie je realizované pomocou kvantizácie priestoru do konečného počtu kvantizačných centier. Časové učenie následne, pomocou metódy *Bayes belief propagation*, vytvorí časovo-priestorovú reprezentáciu vstupných dát. Táto reprezentácia je uložená v stromovej hierarchii *uzlov*, základných výpočtových jednotiek. Toto učenie prebieha v polo-samoorganizujúcom režime. Učiteľ je inherentne prítomný vo vstupných dátach, je to samotný *čas*.

#### 3.1 Architektúra modelu HTM

Väčšina objektov vo svete, napríklad ľudia, autá, internet, ale i reč, majú svoju vnútornú štruktúru (na rozdiel od náhodného šumu). Táto štruktúra je často hierarchická v čase i priestore. HTM sa snaží využiť túto vlastnosť vytváraním podobnej vnútornej reprezentácie dát.

V HTM modeli sú uzly usporiadané do stromovej štruktúry, ktorá sa skladá z  $n_{levs} (\geq 2)$  úrovní  $\mathcal{L}_i$ , očíslovaných indexami od 0 po  $n_{levs} - 1$ . Každá úroveň  $\mathcal{L}_i$  obsahuje niekoľko uzlov  $\mathcal{N}_j^{(i)}$ ,  $j = 1 \dots n_{nodes}^{(i)}$  (pozri 3.1). Uzly v najnižšej úrovni  $\mathcal{L}_0$  sa volajú vstupné alebo *senzory*. Uzly v najvyššej úrovni  $\mathcal{L}_{n_{levs}-1}$  sa volajú výstupné uzly. Zvyšok uzlov, t.j. uzly v úrovňach  $\mathcal{L}_1$  až  $\mathcal{L}_{n_{levs}-2}$ , sa volajú vnútorné uzly.

Každý uzol má prakticky rovnaký učiaci a inferenčný algoritmus. Tento princíp pochádza z kortexu, kde sa kortikálny stĺpec pokladá za základnú výpočtovú jednotku [Mou78]. Hlavný rozdiel v algoritmoch je v senzoroch, keďže pracujú priamo so vstupnými dátami.

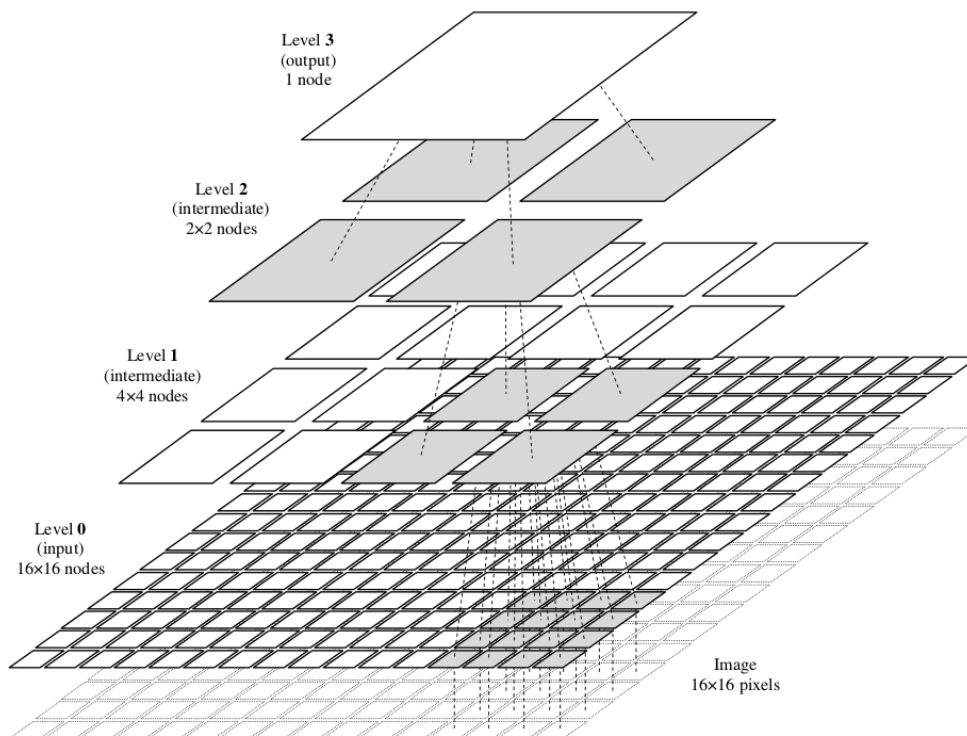
V doméne rozpoznávania vizuálnych vzorov zvyčajne

1. senzory sú vo vzťahu jedna k jednej k obrazovým pixelom,
2. uzly v každej úrovni  $\mathcal{L}_i$  sú usporiadané v obdĺžnikovej mriežke,
3. na výstupnej úrovni je iba jeden výstupný uzol,
4. a uzly v susedných úrovniach (napríklad  $\mathcal{L}_1$  a  $\mathcal{L}_2$ ) nie sú plne prepojené, ale ich prepojenia kopírujú ich *receptívne polia*.

---

<sup>1</sup>Numenta je súkromná americká spoločnosť založená v roku 2005 Jeff Hawkinsom, Dileepom Georgeom a Donnou Dubinskou.

Receptívne polia majú obdĺžnikový tvar a ich veľkosť sa nastavuje podľa počtu uzlov v susedných úrovniach a podľa množstva ich prekrytia tak, aby rovnomerne pokrývali uzly v nižšej úrovni  $\mathcal{L}_{i-1}$ . Vďaka hierarchii, uzly vo vyšších úrovniach pokrývajú, z pohľadu vstupných obrazových dát, väčšiu oblasť.

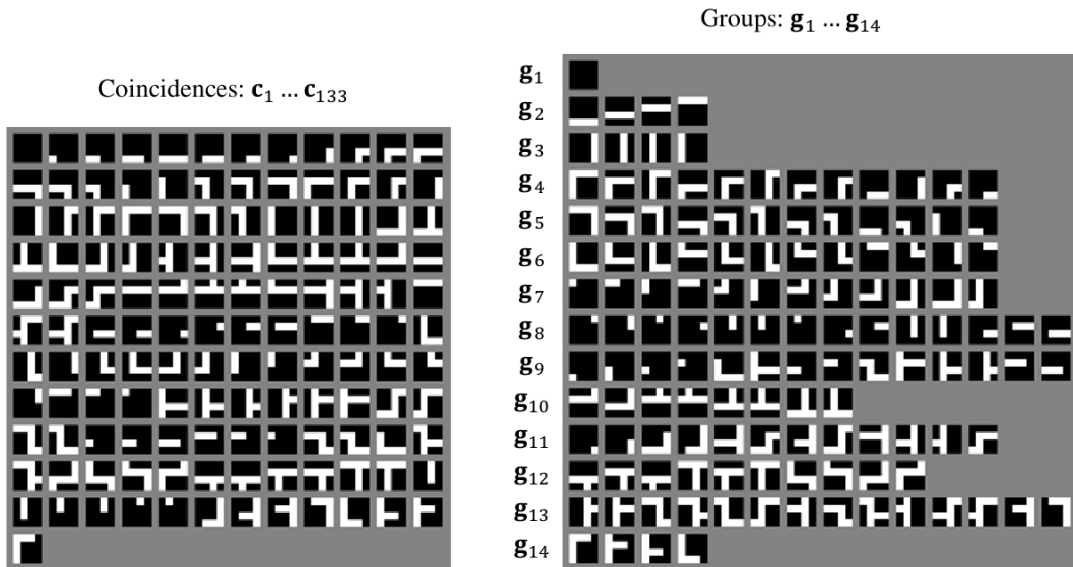


**Obr. 3.1:** Príklad 4-úrovňovej HTM siete. Úroveň 0 má vstupné uzly v mriežke  $16 \times 16$ , kde je každý uzol asociovaný s jedným obrazovým pixelom. Každý uzol na prvej úrovni získava vstup z oblasti  $4 \times 4$  pixelov. Uzly na druhej úrovni získava svoj vstup z  $2 \times 2$  uzlov z predchádzajúcej úrovne, čo je  $8 \times 8$  pixelov zo vstupného obrazu. Jeden výstupný uzol v tretej úrovni pokrýva celý vstupný obraz [Mal11].

## 3.2 Zloženie uzlov

### 3.2.1 Vstupné uzly

Vstupné uzly  $\mathcal{N}^{(0)}$  sú najjednoduchšie uzly, keďže len posúvajú vstupné dáta ďalej do siete podľa aktuálneho prepojenia. Neprebíha v nich učenie ani inferencia, ale môžu sa použiť na extrakciu príznakov, napríklad pomocou Gáborových filtrov.



**Obr. 3.2:** Príklad množiny koincidiencií (vľavo) a časových skupín (vpravo) uzla prvej úrovne po tréningu na obrazoch objektov z jednoduchých čiar. Všetky obrazové výrezy majú  $4 \times 4$  pixelov [Mal11].

### 3.2.2 Vnútorne uzly

Každý vnútorný uzol obsahuje tri základné dátové štruktúry:

1. množinu  $\mathbb{C}$  koincidiencií  $\vec{c}_i$ , kde  $0 < i < n_c$ ,
2. množinu  $\mathbb{G}$  časových skupín  $g_j$ , kde  $0 < j < n_g$ ,
3. a  $n_c \times n_g$  maticu  $\mathbf{PCG}$ ,

ktoré sa počas učenia aktualizujú.

**Koincidencia**  $\vec{c}$  je prototyp vzoru, ktorý reprezentuje podpriestor podobných vzorov. Táto podobnosť sa meria rôzne pre senzory a pre vnútorné uzly. Veľkosť koincidencie závisí od veľkosti receptívneho poľa uzla. V prípade uzlov  $\mathcal{N}^{(1)}$ , koincidencia je výrez vstupného obrázku. V prípade uzlov  $\mathcal{N}^{(i)}$ , kde  $i \geq 2$ , koincidencia  $\vec{c}_i$  môže byť interpretovaná ako vektor príznakov, kedy každý element  $\vec{c}_i[j]$  je index časovej skupiny. Táto časová skupina je vybraná z časových skupín  $\text{child}_j(\mathcal{N})$ , a tak  $|\vec{c}_i| = n_{\text{childs}}(\mathcal{N})$ . Napríklad ak  $\vec{c}_1 = [4, 5, 2]$ , potom  $\vec{c}_1$  vyberá skupinu 4 z hierarchicky nižšieho uzla 1, skupinu 5 uzla 2, a skupinu 2 z uzla 3.

**Časová skupina**  $g$  je množina koincidiencií  $\vec{c}$ , ktoré nie sú nutne podobné v priestorovej doméne, ale mali byť podobné perceptuálne. Napríklad, keď sa vertikálna čiara posunie iba o jeden pixel, jej vektorová reprezentácia sa dramaticky zmení. HTM používa čas na vyriešenie tohto problému. Riešenie je založené na predpoklade, že

vzory prezentované v čase blízko za sebou pravdepodobne pochádzajú z tej istej príčiny, t.j. sú varianty toho istého vzoru.

**PCG** je  $n_c \times n_g$  matica časových prechodov, ktorá ukladá prvorádové časové vzťahy medzi koincenciami. **PCG** matica sa nazýva podľa jej definície  $\mathbf{PCG} = P(\mathbb{C}|\mathbb{G})$ .  $\mathbf{PCG}[i, j] = P(\vec{c}_i|g_j)$  označuje relatívnu pravdepodobnosť výskytu koincencie  $\vec{c}_i$  pri pozorovaní skupiny  $g_j$ .

### 3.2.3 Výstupné uzly

Výstupný uzol môže pracovať priamo ako klasifikátor alebo svoj výstup poskytne na ďalšie spracovanie, či uloženie. Na klasifikáciu je možné použiť prakticky každý klasifikátor (napríklad  $k$ -NN, Support vector machine – SVM, viacvrstvový perceptrón, a pod.), výber závisí od konkrétnej úlohy a vstupných dát.

## 3.3 Učenie

HTM sieť pracuje v dvoch základných režimoch: *učenie* a *inferencia*. Tieto režimy sa nastavujú jednotlivo pre každý uzol, pričom uzly v jednej úrovni pracujú vždy v rovnakom režime.

Cielom učenia je vytvoriť časovo-priestorovú reprezentáciu vstupnej sekvencie dát. Počas učenia uzly najskôr vytvoria priestorovú a následne časovo-priestorovú reprezentáciu dát. Prvá fáza sa nazýva *priestorové učenie*. Druhá fáza sa volá *časové učenie*. Uzly, ktoré ukončili svoje učenie sú prepnuté do inferečného režimu.

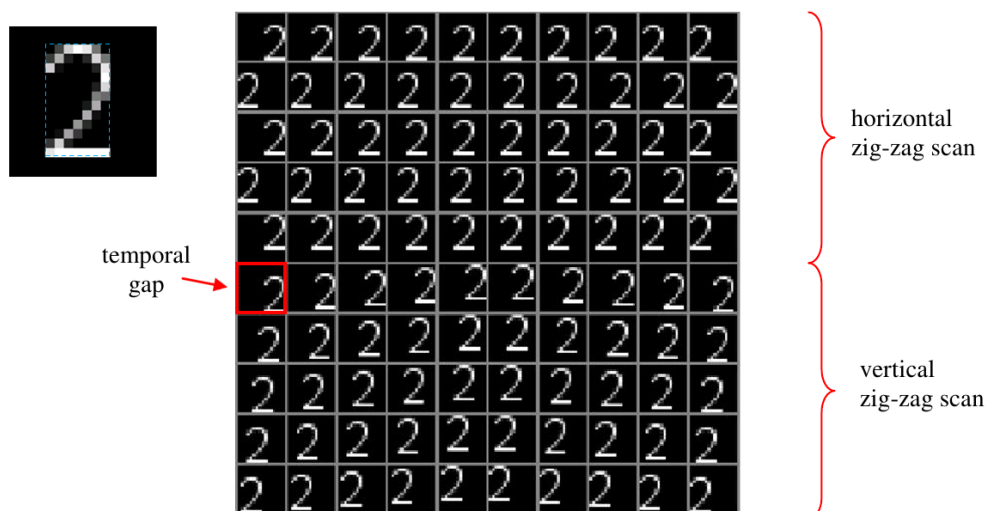
Učenie prebieha bez učiteľa a po jednotlivých úrovniach, začínajúc úrovňou  $\mathcal{L}_1$  a končiac úrovňou  $\mathcal{L}_{n_{levs}-1}$ . Počas učenia úrovne  $\mathcal{L}_i$ , všetky uzly v predchádzajúcich úrovniach už ukončili učenie a pracujú v inferenčnom režime.

Učenie uzlov jednej úrovne môže prebiehať dvomi spôsobmi: (i) každý uzol sa bude učiť samostatne, čím každý dospeje k vlastným naučeným reprezentáciám ( $\mathbb{C}$ ,  $\mathbb{G}$ , a **PCG**), alebo (ii) uzly sa budú učiť spoločne. Pri spoločnom učení sa upravuje iba jedna spoločná reprezentácia dát, ktorá sa po ukončení učenia skopíruje do všetkých uzlov danej úrovne. V prípade rozpoznávania vizuálnych vzorov sa spoločné učenie obyčajne aplikuje na prvých niekoľko úrovni, kde sa učia základne vzory, ako napríklad orientované čiary alebo rohy. Takýto spôsob učenia potom zaručí rovnakú inferenciu nezávisle od pozície uzla v rámci úrovne. Ďalšou výhodou spoločného učenia je menšia výpočtová a pamäťová náročnosť.

### 3.3.1 Trénovacia sekvencia

Na učenie vnútorných uzlov je potrebná trénovacia sekvencia obrazových vzorov. V prípade statických obrazov je potrebné túto sekvenciu vytvoriť, pričom sa musí čo najviac





**Obr. 3.3:** Príklad trénuvacej sekvencie generovanej z jedného obrázku s rozmermi  $16 \times 16$  pixelov (obrázok vľavo hore). Časová medzera je vidieť v mieste, kde sa horizontálne skenovanie spája s vertikálnym. Takého miesta sa označujú značkou, ktorá indikuje časovému učeniu prerušenie prirodzenej sekvencie [Mal11].

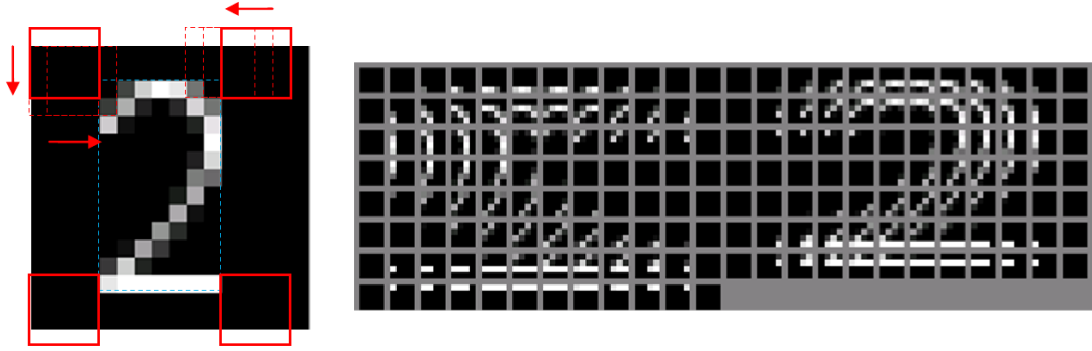
zachovať predpoklad, že vzory blízko seba v čase pochádzajú z toho istého zdroja (príčiny). Algoritmus na vytváranie sekvencií obrazových vzorov by ďalej nemal preferovať určitý smer prechádzania obrazom. Prechádzanie obrazom napríklad horizontálnym smerom nenechá horizontálnu čiaru, a tak sa nepodarí rôzne horizontálne čiary zoskupiť počas časového učenia. Výber vhodného algoritmu je pre správne učenie veľmi dôležitý.

V závislosti od použitia spoločného učenia je možné trénuvacie sekvencie generovať dvomi spôsobmi. Pri zapnutom spoločnom učení sa sekvencia generuje pohybom okna veľkosti receptívneho poľa po vstupnom obraze (pozri obrázok 3.4). Pri vypnutom spoločnom učení sa trénuvací objekt plynulo posúva, mení veľkosť alebo rotuje v rámci obrazu tak, aby bol vždy viditeľný vcelku (pozri obrázok 3.3).

Finálna sekvencia sa zostaví spojením sekvencií z jednotlivých trénuvacích obrazov, pričom sa pre účely časového učenia do sekvencie vložia značky označujúce prechody medzi nesúvisiacimi vzormi (pozri obrázok 3.3). Bez týchto značiek by sa napríklad posledný vzor z jedenej sekvencie mylne pokladal za predchodcu prvého vzoru z nasledujúcej sekvencie.

### 3.3.2 Priestorové učenie

Cielom priestorového učenia je vytvoriť množinu koincidencií, ktoré vhodne reprezentujú vstupné dáta daného uzla.



**Obr. 3.4:** Príklad generovania sekvencie pri učení uzla so zapnutým spoločným učením. Posuvné okno, rovnakej veľkosti ako je receptívne pole uzla, sa pohybuje cez objekt horizontálnym cik-cak spôsobom [Mal11].

V HTM teórii sú opísané tri typy algoritmov priestorového učenia: gaussovský, maticový súčin (angl. dot) a násobenie (angl. product). Gaussovský algoritmus sa používa pre uzly v úrovni  $\mathcal{L}_1$ , ktoré ako vstup dostávajú priamo obrazové výrezy. Vo vyšších úrovniach ( $\geq 2$ ) sa používajú zvyšné dva algoritmy.

Gaussovský algoritmus pracuje počas učenia na princípe online klasterovania s použitím euklidovskej metriky. Aktuálny vstup sa uloží ako nová koincidencia, ak jeho vzdialenosť od ostatných koincidií je väčšia ako parameter *maxDistance*. Niekoľko slabín tohto algoritmu bolo opísaných v [vB10].

V prípade vektorového súčinu a násobenia, daný uzol dostáva na vstup zreťazený výstup uzlov o úroveň nižšie, s ktorými je prepojený. Tento vstup je spracovaný nasledovne:

1. Vo vstupnom vektore od každého uzla z nižšej úrovne sa maximálna hodnota nahradí číslom 1 a zvyšok číslom 0 (vítaz berie všetko). Zreťazený vstupný vektor má nakoniec počet jednotiek zhodný s počtom prepojených uzlov z nižšej úrovne.
2. Výsledný binárny vektor sa pridá do množiny koincidií. Ak sa tam už rovnaký vektor nachádza, zvýši sa počítadlo jeho výskytov o jedna.

### 3.3.3 Časové učenie

Cieľom časového učenia je zachytiť časovú štruktúru vstupných dát. Princíp spočíva v spájaní vzorov, ktoré sa vyskytnú na vstupe v čase blízko za sebou. Dôležitým predpokladom je mať vhodnú vstupnú sekvenciu dát, teda takú, v ktorej sa vizuálny objekt plynulo transformuje v rámci vstupného obrazu.

Pre vytvorenie časových skupín je potrebné zhromaždiť informácie o časovej následnosti vzorov z  $\mathbb{C}$  (koincidií) vo vstupnej sekvencii. Zhromažďovanie prebieha v matici časových následností (angl. Time Adjacency Matrix, skratka TAM), označená  $\mathbf{T}$ .  $\mathbf{T}$  je  $n_c \times n_c$

( $n_c = |\mathbb{C}|$ ) matica, ktorej riadky a stĺpce korešpondujú s koincenciami z  $\mathbb{C}$ . Matica  $\mathbf{T}$  sa používa na konci učenia na získanie množiny  $\mathbb{G}$  a matice **PCG**.

Pre nastavenie maximálnych časových vzdialeností medzi koincenciami, ktoré chceme brať ešte do úvahy, slúži parameter *transitionMemory*. Prírastky v matici  $\mathbf{T}$  sa následne váhujú, a to proporcionálne časovej vzdialenosti. Váha klesá so zväčšujúcou sa časovou vzdialenosťou koincencie  $\vec{c}_{curr}$  a  $\vec{c}_{curr-t}$  vyskytujúcej sa v čase  $t$  krokov do minulosti. Vo výsledku sa  $\mathbf{T}$  aktualizuje nasledovne:

$$\mathbf{T}[\vec{c}_{curr-t}, \vec{c}_{curr}] = \mathbf{T}[\vec{c}_{curr-t}, \vec{c}_{curr}] + (1 + \text{transitionMemory} - t), \quad (3.1)$$

kde  $t \in \{1, \dots, \text{transitionMemory}\}$ .

### 3.3.4 Ukončenie učenia

Keď sa celá vstupná sekvencia vzorov spracuje priestorovým aj časovým učením, nastáva fáza ukončenia učenia, v ktorej sa vykonávajú tieto kroky:

1. Odstránenie zriedkavo sa vyskytujúcich koincencií.
2. Výpočet pravdepodobnosti apriórneho výskytu koincencií.
3. Úprava  $\mathbf{T}$  do symetrického a normalizovaného tvaru.
4. Vytvorenie časových skupín.
5. Výpočet **PCG**.

## 3.4 Inferencia

HTM sieť používa na inferenciu algoritmus Bayesian belief propagation. Bayesian belief propagation vznikol pre potreby inferencie v Bayesových sieťach [Pea88]. Keďže HTM uzol vytvára priestorovo-časovú reprezentáciu dát, bolo treba tento algoritmus upraviť, čo je opísané v [GH09].

Po tom, ako uzly na úrovni  $\mathcal{L}_i$  ukončia svoje učenie, sú prepnuté do inferenčného režimu. Počas inferencie sa naučené reprezentácie uzlov nemenia.

Keď uzol  $\mathcal{N}_j^{(i)}$  dostane na vstup svoje dáta, použije svoj interný model (pamäť) na vytvorenie odpovede, ktorú pošle príslušným uzlom ďalej. Uzol  $\mathcal{N}_j^{(i)}$  vypočíta svoj výstup nasledovným spôsobom:

1. Uzol  $\mathcal{N}_j^{(i)}$  dostane na vstup vektor  $\boldsymbol{\lambda}^- = [\boldsymbol{\lambda}_1^-, \boldsymbol{\lambda}_2^-, \dots, \boldsymbol{\lambda}_m^-]$ , ktorý reprezentuje zretazenie výstupov od prepojených  $m$  uzlov z nižšej úrovne. Každý z jednotlivých výstupov uzlov na nižšej úrovni je vektor  $\boldsymbol{\lambda}^+$ , ktorý obsahuje distribúciu viery v časové skupiny príslušného uzla.

2. Vstup  $\lambda^-$  je najskôr spracovaný algoritmom priestorovej inferencie, ktorý tento vstup porovná s naučenými koincenciami z  $\mathbb{C}$  a vráti vektor, reprezentujúci distribúciu viery v jednotlivé koincencie  $\vec{y}$ .
3. Vektor  $\vec{y}$  je ďalej spracovaný algoritmom časovej inferencie, kde sa vypočíta distribúcia viery v jednotlivé časové skupiny  $\lambda^+$ . Tento vektor je zároveň výstupom celej inferencie. Výpočet zohľadňuje príslušnosť koincencií k časovým skupinám, ktorá je uložená v matici **PCG**.

### 3.4.1 Priestorová inferencia

Cieľom priestorovej inferencie je vypočítať vektor distribúcie vierohodností v koincencie  $\vec{y}$  vzhľadom na aktuálny vstup  $\lambda^-$ . Jednotlivé zložky vektoru  $\vec{y}$  reprezentujú vieru, že vstup  $\lambda^-$  je priestorovo blízko realizáciou koincencie  $\vec{c}_i$ , tj.  $\vec{y}[i] = p(e^-|\vec{c}_i)$ .

#### Gaussovská priestorová inferencia

Tento algoritmus sa používa v úrovni  $\mathcal{L}_1$ , kde je vstupom výrez obrazu. Viera, že  $\lambda^-$  je realizáciou  $\vec{c}_i$ , je daný nasledovne:

$$\vec{y}[i] = \exp \left\{ -\frac{\|\vec{c}_i - \lambda^-\|^2}{\sigma^2} \right\}$$

kde  $\sigma$  určuje správanie, ak sa  $\lambda^-$  vzdaluje od  $\vec{c}_i$ . Ak v dátach nachádza veľa šumu, je vhodné nastaviť parameter  $\sigma$  na vyššiu hodnotu.

#### Priestorová inferencia pomocou maticového súčinu

Maticový súčin sa používa v úrovniach  $\mathcal{L}_i$ , kde  $i \geq 2$ . Vstupom je vektor  $\lambda^-$ . Výstupný vektor  $\vec{y}[i]$  sa vypočíta nasledovne:

$$\vec{y}[i] = p(e^-|\vec{c}_i) = \lambda^- \cdot \vec{c}_i.$$

#### Priestorová inferencia pomocou násobenia

Tento algoritmus sa používa v úrovniach  $\mathcal{L}_i$ , kde  $i \geq 2$ . Vstupom je vektor  $\lambda^-$ . Výstupný vektor  $\vec{y}[i]$  je proporcionálny pravdepodobnosti vzájomného výskytu výstupov jednotlivých prepojených uzlov z nižšej úrovne v kontexte  $\vec{c}_i$ . Pri predpoklade nezávislosti jednotlivých výstupov uzlov, výstupný vektor je daný vzťahom:

$$\vec{y}[i] = p(e^-|\vec{c}_i) = \prod_{j=1}^m p(e^-|\text{child}_j(\mathcal{N}), \vec{c}_i) = \prod_{j=1}^m \lambda_j^-[\vec{c}_i[j]],$$

kde  $\lambda_j^- [\vec{c}_i[j]]$  je zložka vektora na pozícii  $\vec{c}_i[j]$  vo vstupnom vektore  $\lambda_j^-$  z  $\text{child}_j(\mathcal{N})$ .

Takto vypočítaný vektor sa následne normalizuje tak, aby suma jeho zložiek bola rovná práve jednotke. Normalizáciou sa dá vyhnúť výskytu veľmi malých čísiel, pričom tá nemá vplyv na fungovanie HTM siete, keďže neprichádza k zmene relatívnych vzťahov zložiek.

### 3.4.2 Časová inferencia

Cieľom časovej inferencie je vypočítať podmienenú hustotu pravdepodobnosti pre jednotlivé časové skupiny. Tá sa počíta pomocou výstupu priestorovej inferencie,  $\vec{y}$ , napríklad marginalizovaním koincidencií:

$$p(e^-|g_j) = \sum_{i=1}^{n_c} p(e^-|\vec{c}_i, g_j) \cdot P(\vec{c}_i|g_j) = \sum_{i=1}^{n_c} p(e^-|\vec{c}_i) \cdot P(\vec{c}_i|g_j) = \sum_{i=1}^{n_c} \vec{y}[i] \cdot \mathbf{PCG}[i, j],$$

kde sa používa predpoklad, že  $p(e^-|\vec{c}_i, g_j) = p(e^-|\vec{c}_i)$ , pretože znalosť  $g_j$  nie je potrebná pre odhad  $e^-$  v kontexte  $\vec{c}_i$ .

Výstupom časovej inferencie je vektor  $\lambda^+$  uzla  $\mathcal{N}$ . Tento výstup je zložený z podmienených pravdepodobností pre každú z časových skupín:

$$\lambda^+ = p(e^-|\mathcal{N}) = [p(e^-|g_1), p(e^-|g_2), \dots, p(e^-|g_{n_g})].$$

## 4 Prínosy práce

Táto práca sa venuje tematike invariantného rozpoznávania vizuálnych objektov pomocou modelu Hierarchickej časovej pamäte. Konkrétne zameranie je na úlohu klasifikácie vizuálnych objektov umiestnených samostatne na homogénnom pozadí. Invariantné rozpoznávanie sa skúma vzhľadom na posun, zmenu škály a rotáciu objektov.

V tejto časti sú ďalej stručne opísané originálne výsledky dizertačnej práce vzhľadom na stanovené ciele.

### 4.1 Vlastná implementácia HTM algoritmov

Verejne prístupný softvérový balík NuPIC vo verzii 1.7.1 neobsahoval zdrojové kódy hlavných algoritmov HTM modelu (priestorové a časové učenie a inferenciu), preto bolo prvou nevyhnutnou úlohou vytvoriť vlastnú implementáciu týchto algoritmov a tiež pomocných nástrojov.

Implementácia prebehla v programovacom jazyku Python verzii 3.3. Použili sme pri nej knižnice NumPy<sup>2</sup> a PIL<sup>3</sup>. Klasifikačné algoritmy boli použité z balíka scikit-learn<sup>4</sup>.

Táto implementácia obsahuje rovnakú sadu učiacich a inferenčných algoritmov spolu s algoritmi na generovanie sekvencií zo statických obrázkov.

Pred touto implementáciou bola vytvorená ešte jedna jednoúčelová v jazyku MATLAB, ktorej cieľom bolo otestovať nový algoritmus na generovanie sekvencií (pozri sekciu 4.5).

## 4.2 Metodologické porovnanie HTM a modelu okamžitého vizuálneho rozpoznávania

Prvým výskumným prínosom práce je metodologické porovnanie modelu HTM a Serreho modelu okamžitého rozpoznávania objektov (angl. Immediate object recognition, skratka IVR), ktorý vychádza z modelu HMAX.

Oba modely sú hierarchické, predpokladajú vstup vo forme sekvencie a biely šum. Na druhej strane, IVR nepoužíva pravdepodobnostný framework, vyžaduje ručne zadané dáta a prepojenia (učí sa iba selektivita S jednotiek od druhej vrstvy vyššie) a vyžaduje kategorizované vstupné dáta, keďže priamo obsahuje klasifikátor.

## 4.3 Meranie invariance HTM

Dôležitým výskumným prínosom práce je meranie zložitosti klasifikácie obrazových dát s použitím HTM a porovnanie získaných výsledkov s výsledkami merania vo vstupnom priestore. Takéto meranie zložitosti môže prebiehať rôznymi spôsobmi, napríklad meraním klasifikačnej úspešnosti. V tejto práci sme sa rozhodli použiť všeobecnejší prístup a merať určité charakteristiky priestoru, ktoré rôzne opisujú rozloženie dát vzhľadom na úlohu klasifikácie.

Obrazové dáta obsahujú vizuálne objekty, ktoré boli rôzne posunuté, pootočené a škálované. Meraním klasifikačnej zložitosti takýchto dát je možné analyzovať, či a ako vie model HTM prispieť k väčšej invariantnosti voči spomínaným transformáciám.

### 4.3.1 Meranie klasifikačnej komplexity

Podľa [BH06] zložitost klasifikácie pochádza z troch hlavných zdrojoch: nejednakočetnosť tried, komplexita hranice medzi triedami a nedostatok dát pri väčšej dimenzionalite dát.

Konkrétne, Ho a Basu v [HB02] navrhli sadu meraní, ktoré sa zameriavajú na druhý zdroj komplexity klasifikácie – komplexitu hranice medzi triedami. Táto množina kritérií sa dá rozdeliť do niekoľko skupín:

---

<sup>2</sup><http://www.numpy.org>

<sup>3</sup><http://www.pythonware.com/products/pil/>

<sup>4</sup><http://scikit-learn.org/>

**Meranie prekrytia hodnôt atribútov medzi triedami.** Tieto merania sa snažia zachytiť ako dobre môžu atribúty separovať príklady z rôznych tried a zahŕňajú: maximálny Fišerov diskriminačný faktor (F1), prekrytie boxov ohraničujúcich triedy (F2) a maximálnu efektívnosť atribútu (F3).

**Meranie separability tried.** Tieto merania sa zameriavajú na charakterizovanie separability tried z pohľadu zložitosti hranice medzi triedami a zahŕňajú: minimálnu sumu štvorcov chybných vzdialeností lineárneho klasifikátora (L1), trénovaciu chybu lineárneho klasifikátora (L2), podiel príkladov na hranici tried (N1), podiel priemerných vzdialeností v rámci triedy a mimo nej (N2) a leave-one-out chybu 1-NN klasifikátora (N3).

**Meranie geometrie, topológie a hustoty podpriestorov.** Tieto merania poskytujú iba nepriamy odhad separability tried. Predpokladajú, že klasifikačný problém pozostáva z niekoľkých podpriestorov, ktoré definujú jednotlivé triedy. Tvar, pozícia a prekryv týchto podpriestorov následne vypovedá o tom, ako dobre sa dajú jednotlivé triedy separovať. Tieto merania obsahujú: nelinearitu lineárneho klasifikátora (L3), nelinearitu 1-NN klasifikátora (N4), podiel maximálne pokrývajúcich gulí (T1) a priemerný počet príkladov na dimenziu (T2).

Výsledky pre 2-úrovňovú HTM sieť

Transformácia	Priestor	F1	F2	F3	F4	L1	L2	N1	N2	N3	L3	N4	T1	T2
-	orig	28.965	0.000	0.921	0.959	0.143	0.010	0.062	0.712	0.042	0.229	0.000	0.993	0.047
	HTM	172.478	0.000	0.978	0.993	0.056	0.021	0.062	0.454	0.042	0.500	0.005	0.953	0.800
rot	orig	15.753	0.000	0.808	0.980	0.491	0.000	0.014	0.552	0.000	0.000	0.000	0.994	0.281
	HTM	2.637	0.000	0.911	0.994	0.082	0.021	0.025	0.395	0.008	0.500	0.035	0.976	4.800
	HTM	1.750	0.000	0.669	0.993	0.058	0.021	0.044	0.374	0.025	0.491	0.025	0.970	8.000
škála	orig	11.044	0.000	0.743	0.989	0.535	0.008	0.037	0.670	0.020	0.158	0.023	0.996	0.422
	HTM	1.469	0.000	0.771	0.996	0.060	0.021	0.042	0.364	0.024	0.500	0.239	0.969	7.200

Výsledky pre 3-úrovňovú HTM sieť

Transformácia	Priestor	F1	F2	F3	F4	L1	L2	N1	N2	N3	L3	N4	T1	T2
-	orig	28.965	0.000	0.921	0.959	0.143	0.010	0.062	0.712	0.042	0.229	0.000	0.993	0.047
	HTM	39.733	0.000	0.979	0.994	0.045	0.021	0.062	0.414	0.042	0.500	0.018	0.879	2.400
rot	orig	15.753	0.000	0.808	0.980	0.491	0.000	0.014	0.552	0.000	0.000	0.000	0.994	0.281
	HTM	2.306	0.000	0.879	0.996	0.048	0.021	0.040	0.325	0.022	0.500	0.137	0.944	14.400
trans	orig	3.540	0.000	0.601	0.986	0.305	0.050	0.530	0.031	0.114	0.344	0.114	0.983	4.235
	HTM	0.688	0.000	0.653	0.993	0.045	0.021	0.048	0.241	0.031	0.500	0.309	0.959	24.000
škála	orig	11.044	0.000	0.743	0.989	0.535	0.008	0.037	0.670	0.020	0.158	0.023	0.996	0.422
	HTM	1.261	0.000	0.906	0.994	0.045	0.021	0.056	0.352	0.036	0.500	0.258	0.910	7.200

**Tabuľka 1:** Výsledky merania klasifikačnej komplexity dát v originálnom aj HTM priestore 2-úrovňovej HTM siete (tabuľka vyššie) a 3-úrovňovej HTM siete (tabuľka nižšie). Výsledky sú aritmetickým priemerom príslušných meraní pre všetkých 48 skupín dát s dvomi triedami vygenerovaných z pôvodných dát pomocou prístupu “jedna trieda voči zvyšku”. Skratka “rot” znamená rotácia a “trans” translácia v horizontálnom smere.



## 4.4 Porovnanie modelu HTM s modelom DBN

Ďalším prínosom práce je metodologické a experimentálne porovnanie modelu HTM a modelu Deep Belief Network (DBN).

Z hľadiska metodologického porovnania je jeden z najväčších rozdielov v tom, že model DBN nepracuje s časovou doménou vstupných dát. Oba modely sa radia do skupiny s tzv. hlbokou architektúrou [Ben09, ARK10]. Model HTM, vďaka možnosti prekryvu receptívnych polí či zdieľanému učeniu, má dobrý predpoklad pre pozitívnu invarianciu, model DBN takúto inherentnú vlastnosť nemá.

Experimentálne porovnanie sa realizovalo na klasifikačnej úlohe pomocou databázy obrazov listov stromov [Hal12] a s použitím  $k$ -NN klasifikátora. Databáza obsahuje 11 kategórií, každá má 5 šedoškálových obrazov rozmeru  $64 \times 64$  pixelov. Porovnanie prebiehalo v dvoch experimentoch – v prvom sa testovala invariancia voči rotácii (celkovo 660 obrazov) a druhom zasa invariancia voči zmene škály (celkovo 275 obrazov). Pri oboch experimentoch sa menil podiel trénovacích obrazov, a to od 20 do 50 percent. Pri nastavení parametrov oboch modelov sme dbali na to, aby mali porovnateľné podmienky pre učenie.

### 4.4.1 Výsledky a závery

Trn:tst	k-NN	DBN & k-NN	HTM & k-NN
20:80	34.54 (1.52)	56.07 (2.41)	63.53 (1.91)
30:70	37.75 (1.30)	60.92 (3.04)	69.90 (1.99)
40:60	38.33 (1.25)	64.10 (2.32)	73.35 (2.90)
50:50	38.82 (1.09)	66.76 (2.91)	75.36 (2.46)

**Tabuľka 2:** Priemerné klasifikačné úspešnosti v % a príslušné štandardné odchýlky získané z prvého experimentu (invariancia voči rotácii) pre rôzne pomery počtov trénovacích a testovacích dát.

Trn:tst	k-NN	DBN & k-NN	HTM & k-NN
20:80	17.75 (2.13)	39.43 (5.51)	57.31 (3.81)
30:70	17.95 (1.19)	50.67 (4.85)	61.34 (3.53)
40:60	19.79 (2.19)	64.18 (2.14)	65.39 (3.87)
50:50	20.10 (1.77)	58.80 (3.48)	66.44 (3.48)

**Tabuľka 3:** Priemerné klasifikačné úspešnosti v % a príslušné štandardné odchýlky získané z druhého experimentu (invariancia voči škále) pre rôzne pomery počtov trénovacích a testovacích dát.

Celkové výsledky sú uvedené v tabuľkách 2 a 3. Klasifikačná úspešnosť bola analyzovaná z pohľadu rôznych pomerov trénovacích a testovacích dát. Pre lepšie porovnanie

uvádzame aj klasifikačnú úspešnosť priamo vo vstupnom priestore. Pre každú kombináciu sme vykonali 20 behov a uvádzame ich priemery a štandardné odchýlky.

Na základe výsledkov sme dospeli k nasledovným záverom:

- oba modely dosiahli klasifikačnú úspešnosť významne vyššiu v porovnaní s pôvodným priestorom,
- klasifikačná úspešnosť sa zvyšovala so zvyšujúcim sa počtom tréningových obrazov, v prípade prvého experimentu sa klasifikačná úspešnosť modelu DBN zvýšila o 10,69 percentuálnych bodov, zatiaľ čo v prípade modelu HTM to bolo o 11,83 percentuálnych bodov,
- a tiež, že v prípade oboch experimentov, model HTM dosiahol vyššiu klasifikačnú úspešnosť ako model DBN.

## 4.5 Nový spôsob generovania sekvencie vzorov pre časové učenie

Jedným z hlavných prínosov tejto práce je navrhnutie a otestovanie nového algoritmu na generovanie sekvencií zo statických obrazov pre časové učenie, ktorý sme nazvali *pair-wise explorer*.

### 4.5.1 Algoritmus generovania sekvencií

Navrhnutý algoritmus je možné sumarizovať v troch hlavných krokoch:

1. Náhodný výber informačne hodnotného vzoru  $P^F$  zo vstupného obrazu.
2. Náhodný výber druhého vzoru  $P^S$  s pravdepodobnosťou proporcionálnou euklidovskej vzdialenosti stredov oboch vzorov.
3. Vytvorenie minisekvencie dvoch vzorov, ku ktorej sa pridá značka, signalizujúca samostatnú sekvenciu.

Tieto kroky sa opakujú, kým sa nevytvorí žiadaný počet párov, alebo kým sa nespracujú všetky obrazy. Matica TAM môže byť aktualizovaná s jednotkovým prírastkom na miestach zodpovedajúcich získaným párom vzorov. Týmto spôsobom by väčšina aktualizácií mala byť pre páry vzorov s vysokou časovou koherenciou.

**Krok 1.** Vybrať informačne hodnotný vzor nie je jednoduchý problém a je mu venovaná nemalá výskumná snaha. Aj napriek tomu existujú algoritmy, ktoré sa dajú v danej situácii aplikovať. Jedným zo všeobecnejších algoritmov je Metropolis-Hastings (M-H) algoritmus [MRR<sup>+</sup>53, Has70]. Patrí do skupiny “Markov chain Monte Carlo” metód na generovanie vzoriek (stavov)  $\mathbf{s}$  z akejkoľvek pravdepodobnostnej distribúcie  $P(\mathbf{s})$ . Jedinou podmienkou je definovanie pravdepodobnostnej funkcie  $L(\mathbf{s})$  proporcionálnej želanej pravdepodobnostnej distribúcie  $P(\mathbf{s})$  a návrhovej funkcie, ktorá definuje pravdepodobnosť, že navrhnutá vzorka  $\mathbf{s}'$  nasleduje po aktuálnej vzorke  $\mathbf{s}^t$ , formálne  $Q(\mathbf{s}'|\mathbf{s}^t)$ . Pravdepodobnostná funkcia  $L(\mathbf{s})$  by mala reflektovať informačnú hodnotu príslušného

vzoru. Viac informácií o aplikovaní tohto algoritmu na úlohu rozpoznávania vizuálnych objektov je uvedených v článku v [ŠBVŠ12].

**Krok 2.** Cieľom tohto kroku je vybrať druhý vzor  $P_S$  z pravdepodobnostnej distribúcie  $P(D)$  ( $D$  označuje množinu všetkých vzdialeností v priestore koordinát), ktorá odráža časovú koherenciu medzi párami vzorov. Pozícia vzoru je daná dvojicou  $(d, \theta)$ , kde  $d$  je vzdialenosť od centra  $P^F$  a  $\theta$  je ich vzájomný uhol vzhľadom na os  $x$ . Jednou z možností je použiť pravdepodobnostnú distribúciu  $P(D)$ , ktorá lineárne závisí od pôvodného pravidla (pozri 3.1):

$$P(D = d) = \begin{cases} \frac{2(d_{max} - d + 1)}{d_{max}(d_{max} + 1)} & \text{if } d \in \{1 \dots d_{max}\} \\ 0 & \text{otherwise} \end{cases}, \quad (4.1)$$

kde  $d_{max}$  je parameter, ktorý definuje maximálnu vzdialenosť od centra  $P_F$ . Tento parameter by mal byť nastavený proporčne k veľkosti receptívneho poľa uzla.

Uhol  $\theta$  je náhodne vybraný rovnomerne zo všetkých možných uhlov (meraný v radiánoch). Finálna pozícia centra druhého vzoru  $P^S$  je daná vzťahmi:

$$\begin{aligned} P_x^S &= P_x^F + d \cos(\theta) \\ P_y^S &= P_y^F + d \sin(\theta). \end{aligned}$$

**Krok 3.** V tomto kroku sa vyberie z obrazu pár vzorov  $(P^F, P^S)$  a vytvorí sa z nich minisekvencia. Pre správne fungovanie časového učenia je na koniec sekvencie pridaná značka, ktorá signalizuje koniec minisekvencie (pozri sekciu 3.3.3), aby pri spojení s nasledujúcim párom neprišlo k nesprávnemu spojeniu vzorov medzi rôznymi minisekvenciami.

### Použitie algoritmu pre viacúrovňovú HTM sieť

Použitie navrhnutého algoritmu pre HTM sieť s viac ako jednou vnútornou úrovňou je priamočiare. Pre generovanie trénovacej sekvencie na časové učenie vnútorných uzlov v takýchto úrovniach je potrebné zvýšiť hodnotu maximálnej vzdialenosti  $d_{max}$  v druhom kroku algoritmu. Zvýšenie tejto hodnoty by malo zodpovedať zväčšeniu receptívnych polí daných uzlov (zväčšenie sa myslí vzhľadom na vstupný obraz). Týmto spôsobom môžu uzly spracovať informácie na dlhšej časovej škále, ktorá zodpovedá väčšej vzdialenosti vo vstupnom obraze, aj z takto krátkej sekvencie.

### 4.5.2 Experimenty

Algoritmus bol experimentálne porovnaný s pôvodným algoritmom, pripomínajúcim Brownov pohyb, na databáze jednoduchých geometrických objektov (obdĺžniky, trojuholníky a kružnice) umiestnených jednotlivo na homogénnom pozadí. Objekty sú na obrazoch, veľkosti  $64 \times 64$  pixelov, náhodne posunuté, pootočené a škálované, vždy je však na obrázku celý objekt. Viac informácií o dátach je uvedených v článku [ŠBVŠ12].

Počet tré. obr.	Orig. priestor	Priestorov špec.			Bag-of-features		
		PU	OE	PE	PU	OE	PE
10	37.1	36.3	42.1	42.8	40.3	65.2	66.9
20	39.8	39.1	46.8	47.6	43.0	72.8	74.3
40	44.1	42.9	53.7	54.9	46.7	78.5	79.7
60	48.1	45.3	58.7	59.1	48.6	80.7	82.2
80	50.8	47.4	63.6	65.6	50.3	82.6	83.0
100	53.7	49.5	66.9	69.6	52.3	84.4	84.6
150	59.6	53.2	72.9	74.2	55.4	86.0	87.0
200	63.7	56.5	76.2	77.6	56.9	86.8	88.7
250	67.2	59.1	79.2	80.2	59.1	87.8	89.0
300	69.9	61.5	82.2	82.4	60.7	88.5	89.6

**Tabuľka 4:** Prehľad hodnôt klasifikačnej úspešnosti (v %) získaných pomocou  $k$ -NN klasifikátora v rôznych priestoroch a pri rôznom počte tréningových obrazov (prvý stĺpec zľava) pre najvyšší počet aktualizácií matice TAM (32 768). Ďalšie stĺpce obsahujú výsledky pre priestorové učenie (PU), originálny explorer (OE) a “pair-wise” explorer (PE) pre priestorovo špecifickú variantu a pre “bag-of-features” variantu.

HTM sieť pozostávala z jednej úrovne uzlov, ktoré sa učili zdieľaným učením. Receptívne pole každého uzla bolo  $8 \times 8$  pixelov. Vzhľadom na charakter dát boli zvolené nasledovné hodnoty voliteľných parametrov siete HTM: počet koinციencií 512, počet časových skupín 64 a parameter *transitionMemory* 4.

S prihliadnutím na charakter dát bola v 1. kroku pair-wise algoritmu definovaná funkcia  $L(\cdot)$  nasledovne:

$$L(\mathbf{s}) = E(\mathbf{s}^2)^l = E([P_{x,y}^{(i)}]^2)^l, \quad (4.2)$$

kde  $P_{x,y}^{(i)}$  je vzor so stredom na koordinátach  $(x, y)$  tréningového obrazu  $i^{th}$ ,  $E(\cdot)$  je aritmetický priemer zložiek vzoru a  $l$  kladná celočíselná konštanta, v našom prípade sme použili  $l = 4$ .

Výstup HTM siete sme sa rozhodli vytvoriť v dvoch variantoch. V prvom variante výstup pozostával zo zrefazenia výstupov jednotlivých uzlov (priestorovo špecifický výstup). V druhom variante boli výstupy jednotlivých uzlov po zložkách sčítané (“bag-of-features” výstup).

### 4.5.3 Výsledky a závery

Na základe výsledkov uvedených v tabuľke 4 a na obrázku 4.1 možno konštatovať, že “pair-wise” explorer viditeľne zlepšil rýchlosť konvergence k teoretickému maximu klasifikačnej úspešnosti vzhľadom na dĺžku tréningovej sekvencie a tiež vzhľadom na počet tréningových obrázkov. Najvyšší relatívny nárast klasifikačnej úspešnosti bol nameraný pre nižšie hodnoty počtu aktualizácií matice TAM. Konkrétne, pre 2048 aktualizácií matice TAM použitie “pair-wise” algoritmu zvýšilo klasifikačnú úspešnosť približne o 7,5 percenta oproti pôvodnému prístupu.

## 5 Zhrnutie prínosov práce

Táto práca sa venuje tematike invariantného rozpoznávania vizuálnych objektov pomocou modelu Hierarchickej časovej pamäte (anglicky Hierarchical Temporal Memory, skráteno HTM). Konkrétne zameranie je na úlohu klasifikácie vizuálnych objektov umiestnených samostatne na homogénnom pozadí. Invariantné rozpoznávanie je skúmané vzhľadom na posun, zmenu škály a rotáciu objektov.

V nasledujúcej časti sú zhrnuté prínosy práce vzhľadom na jednotlivé stanovené ciele.

### **Vývoj vlastnej implementácie HTM modelu a podporných nástrojov pre umožnenie vykonania experimentov a zmien algoritmov**

Softvérový balík NuPIC vo verzii 1.7.1 neobsahoval zdrojové kódy hlavných algoritmov HTM modelu (priestorové a časové učenie a inferenciu), preto bola vytvorená vlastná implementácia týchto algoritmov, ako aj pomocných nástrojov v programovacom jazyku Python verzii 3.3.

### **Porovnanie modelu HTM s inými aktuálnymi bioinšpirovanými modelmi určenými pre rozpoznávanie obrazov**

Prvým výskumným prínosom práce je metodologické porovnanie modelu HTM a Serreho modelu okamžitého rozpoznávania objektov (Immediate object recognition), ktorý vychádza z modelu HMAX. Modely sú porovnané z hľadiska ich predpokladov, architektúry, spôsobu učenia a možností klasifikácie.

Ďalším výskumným prínosom práce je metodologické a experimentálne porovnanie modelu HTM s modelom Deep Belief Network (DBN). Experimentálne porovnanie sa realizovalo pomocou klasifikačnej úlohy aplikovanej na obrazovú databázu listov stromov a s použitím  $k$ -NN klasifikátora. Porovnanie prebiehalo v dvoch experimentoch – prvý testoval invarianciu voči rotácii a druhý testoval invarianciu voči zmene škály. Pri oboch experimentoch sa menil podiel trénovacích obrázkov v rozsahu od 20 do 50 percent. Pri nastavovaní parametrov oboch modelov sme dbali na to, aby boli podmienky pre učenie porovnateľné. Výsledky ukázali pozitívnu koreláciu medzi množstvom trénovacích obrazov a klasifikačnou úspešnosťou. V prípade prvého experimentu bolo zlepšenie modelu DBN medzi najnižším a najvyšším množstvom trénovacích obrazov o 10,69 percentuálnych bodov a modelu HTM o 11,83 percentuálnych bodov.

### **Výskum ako je model HTM schopný zlepšiť invariantné vlastnosti vstupných obrazových dát**

Dôležitým výskumným prínosom práce je tiež odhad zložitosti klasifikácie na obrazovej databáze Pictures, ktorá je súčasťou balíka NuPIC. HTM sieť bola trénovaná na podmnožine dát z tejto databázy, nazvanej “canonical”, ktorá obsahuje 48 kategórií obrazov,

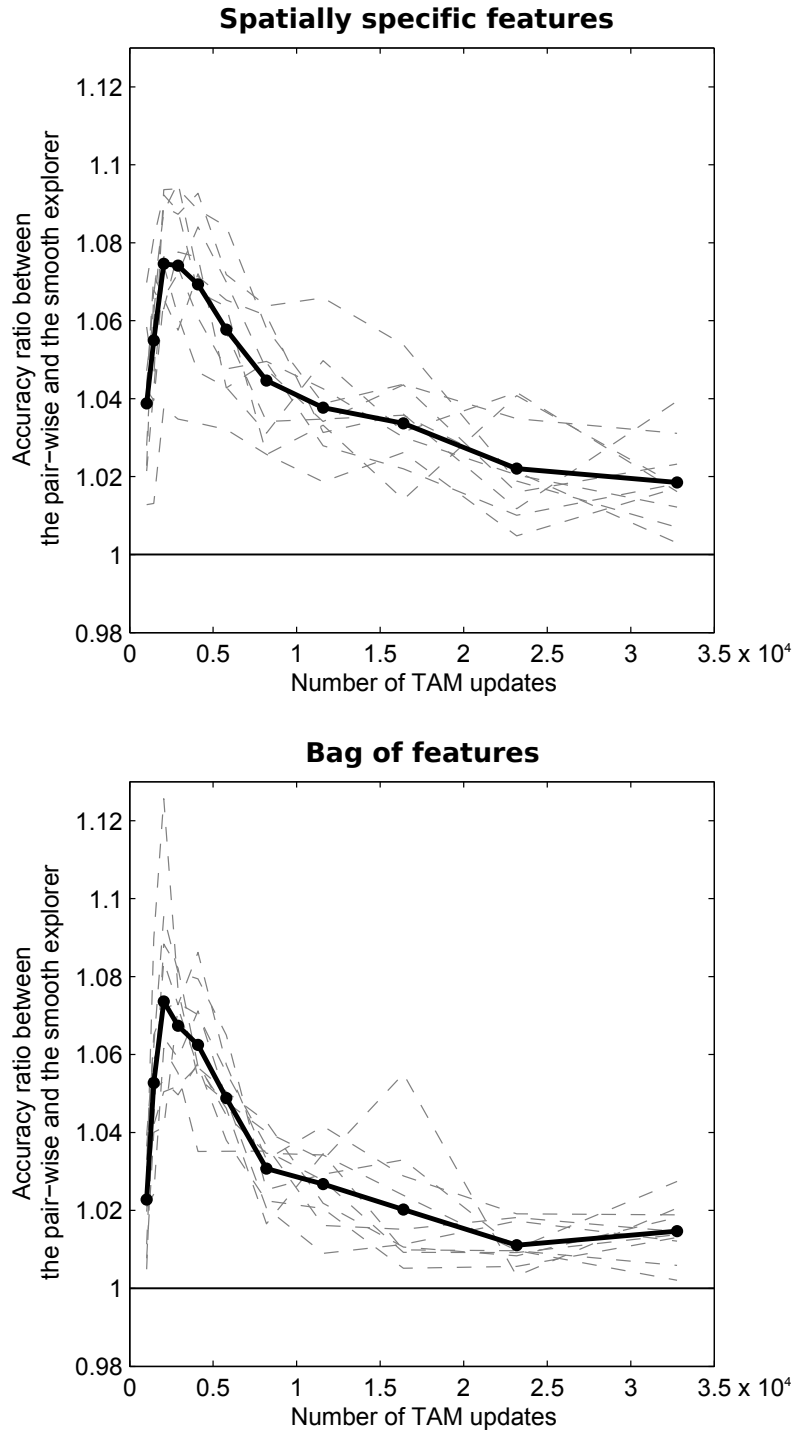
pre každú kategóriu iba jeden príklad. Tieto obrazy boli následne nezávisle posunuté, pootočené a škálované, na základe čoho vznikli tri nové skupiny obrazov. Na originálnych obrazoch a tiež na troch nových skupinách obrazov, v pôvodnom aj HTM priestore, bolo aplikovaných 13 mier zložitosti klasifikácie. Získané výsledky ukazujú, že pre všetky skupiny obrazov boli hodnoty väčšiny mier pre HTM priestor lepšie v porovnaní s pôvodným priestorom.

### **Štúdium spätného šírenia informácií v modeli HTM pre použitie na rozpoznávanie vizuálnych obrazov**

Boli vykonané prvotné experimenty na úlohách segmentácie a rekoštrukcie vizuálnych objektov, ale pre získanie publikovateľných výsledkov je potrebný ešte ďalší výskum.

### **Vývoj nového algoritmu pre generovanie tréningových sekvencií zo statických obrazov**

Jedným z hlavných prínosov práce je navrhnutie a otestovanie nového algoritmu na generovanie sekvencií zo statických obrazov pre časové učenie, ktorý sme nazvali *pair-wise explorer*. Nový algoritmus je určený pre jedno aj viacvrstvovú architektúru HTM siete. Algoritmus je experimentálne porovnaný s pôvodným algoritmom, pripomínajúcim Brownov pohyb, na databáze jednoduchých geometrických objektov (obdĺžniky, trojuholníky a kružnice) umiestnených jednotlivo na homogénnom pozadí. Objekty sú na obrazoch náhodne posunuté, pootočené a škálované, vždy je však na obrázku celý objekt. Na základe výsledkov možno konštatovať, že *pair-wise explorer* viditeľne zlepšil rýchlosť konvergencie k teoretickému maximu klasifikačnej úspešnosti vzhľadom na dĺžku tréningovej sekvencie a tiež vzhľadom na počet tréningových obrazov. Najvyšší relatívny nárast klasifikačnej úspešnosti bol nameraný pre nižšie hodnoty počtu aktualizácií matice časových závislostí (TAM). Konkrétne, pre 2048 aktualizácií TAM matice použitie *pair-wise* algoritmu zvýšilo klasifikačnú úspešnosť približne o 7,5 % oproti pôvodnému prístupu.



**Obr. 4.1:** Grafy zobrazujú priebeh klasifikačnej úspešnosti pre rôzne počty aktualizácii TAM matice a pre obe varianty výstupu siete (horný graf zobrazuje variantu s priestorovo špecifickým výstupom a spodný graf variantu bag-of-features). Prerušované čiary reprezentujú behy s rôznym počtom tréningových obrazov, plná čiara zobrazuje ich aritmetický priemer. Najväčší nárast klasifikačnej úspešnosti pre “pair-wise” explorer, oproti originálnemu exploreru, sme zaznamenali pri 2048 aktualizáciach TAM matice.

## 6 Summary

This work addresses the use of bio-inspired machine learning model, called Hierarchical Temporal Memory (HTM), in the context of invariant visual recognition. In particular, the focus was given to classification of individual objects on homogeneous backgrounds subject to three types of transformation: translation, rotation, and scaling.

In the introduction, the necessary background from neuroscience and visual object recognition is provided. The HTM model is described in detail and put in the context of other similar state-of-the-art models.

The Numenta platform for intelligent computation (NuPIC) software package, in its latest version 1.7.1, contained only binaries of the core algorithms (i.e., spatial and temporal poolers), thus a custom implementation has been developed to enable changes in those algorithms. The Python programming language in version 3.3 has been used.

Comparison of the HTM model with the Serre’s model of Immediate object recognition, the successor of HMAX model, is provided. The similarities and differences are described in terms of their assumptions, architecture, learning, and classification capabilities.

Estimation of the data classification complexity on Pictures dataset (available in the NuPIC package) is performed. The HTM was trained on a “canonical” subset of the dataset containing 48 categories, one image per category. These images were then translated, rotated, and scaled independently, forming three new extended datasets. A set of 13 data classification complexity measures were computed on the original dataset, that was used for training, as well as on the three new datasets both for in the original space and HTM space. The results show that the measures yield better results for the HTM feature space compared with the original space for all the datasets.

Next section contains comparison of the HTM model with the DBN model. Both models were evaluated also on an exemplary classification task: categorization of tree leaves using  $k$ -NN classifier. The evaluation was performed in two experiments – one to test the rotational invariance and one to test the scale invariance in relation to the training set size. The number of training examples varied from 20 % to 50 %. The improvement of classification accuracy (CA) for increased ratio of training images was apparent; in the case of the Experiment 1, CA of DBN improved by 10.69 percentage points while CA of HTM improved even more by 11.83 percentage points.

A novel image explorer, called the *pair-wise explorer*, has been proposed and experimentally analyzed. The algorithm is described for the use for HTM networks with one or more intermediate levels. The novel algorithm has been compared with the Brownian-like smooth algorithm on a dataset of simple geometric objects (i.e., rectangles, triangles, and circles) placed on homogeneous background. The objects were randomly translated, rotated, and scaled. Based on the obtained results, we conclude that using the pair-wise explorer significantly increased the speed of convergence to the theoretical maximum of CA with respect to both the length of the training sequence and the number of training images. The highest relative increase of CA was measured for lower numbers of Time



adjacency matrix (TAM) updates. In particular, for 2048 TAM updates the pair-wise explorer yield about 7.5 % better CA than the original explorer.

## Literatúra

- [ARK10] I. Arel, D. Rose, and T. Karnowski. Deep machine learning - a new frontier in artificial intelligence research. *IEEE Computational Intelligence Magazine*, 5: 13–18, 11/2010 2010.
- [Ben09] Y. Bengio. Learning Deep Architectures for AI. *Foundations and Trends in Machine Learning*, 2(1): 1–127, 2009.
- [BH06] M. Basu and T. K. Ho. *Data complexity in pattern recognition*. Springer, 2006.
- [ByRKT<sup>+</sup>98] P. Bach-y Rita, K. A. Kaczmarek, M. E. Tyler, J. Garcia-Lara, et al. Form perception with a 49-point electro tactile stimulus array on the tongue: A technical note. *Journal of rehabilitation research and development*, 35: 427–430, 1998.
- [Cre77] O. D. Creutzfeldt. Generality of the functional structure of the neocortex. *Naturwissenschaften*, 64(10): 507–517, 1977.
- [FVE91] D. J. Felleman and D. C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex*, 1(1): 1–47, 1991.
- [GH05] D. George and J. Hawkins. A hierarchical Bayesian model of invariant pattern recognition in the visual cortex. In D Prokhorov, editor, *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, volume 3, pages 1812–1817. IEEE, 2005.
- [GH09] D. George and J. Hawkins. Towards a mathematical theory of cortical micro-circuits. *PLoS Computational Biology*, 5(10), 2009.
- [Hal12] M. Halabuk. Testing the hierarchical neural network DBN in invariant object recognition. Master’s thesis, Comenius University in Bratislava, Slovakia, 2012.
- [Has70] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57: 97–109, 1970.
- [HB02] T. K. Ho and M. Basu. Complexity measures of supervised classification problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3): 289–300, 2002.
- [HB04] J. Hawkins and S Blakeslee. *On intelligence*. Henry Holt and Company, New York, 2004.
- [Mal11] D. Maltoni. Pattern Recognition by Hierarchical Temporal Memory. Technical report, University of Bologna, April 2011.
- [Mou78] V. B. Mountcastle. *An organizing principle for cerebral function: the unit model and the distributed system*. MIT Press, 1978.

- [MRR<sup>+</sup>53] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A.H. Teller, E. Teller, et al. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21: 1087–1092, 1953.
- [Pea88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, volume 88. Morgan Kaufmann, 1988.
- [RPHS90] A. W. Roe, S. L. Pallas, J. O. Hahm, and M. Sur. A map of visual space induced in primary auditory cortex. *Science*, 250(4982): 818–820, 1990.
- [ŠBVŠ12] S. Štolc, I. Bajla, K. Valentín, and R. Škoviera. Pair-wise temporal pooling method for rapid training of the htm networks used in computer vision applications. *Computing and Informatics*, 31(4): 901–919, 2012.
- [vB10] S. Štolc and I. Bajla. On the Optimum Architecture of the Biologically Inspired Hierarchical Temporal Memory Model Applied to the Hand-Written Digit Recognition. *Measurement Science Review*, 10(2): 28–49, 2010.

## Publikácie autora

- [1] S. Štolc, I. Bajla, K. Valentín, and R. Škoviera. Temporal Pooling Method for Rapid HTM Learning Applied to Geometric Object Recognition. In *Proceedings of the 8th International Conference on Measurement*, pages 59–64, 2011.
- [2] K. Valentín. Hierarchical Temporal Memory – What is New? Poster at The 7th International Summer School on Pattern Recognition, 2011. Plymouth, UK.
- [3] S. Štolc, I. Bajla, K. Valentín, and R. Škoviera. Pair-wise temporal pooling method for rapid training of the HTM networks used in computer vision applications. *Computing and Informatics*, 32:1001–1019, 2012.
- [4] R. Škoviera, K. Valentín, S. Štolc, and I. Bajla. Detekcia anomálneho správania biologicky inšpirovanou sieťou. Technical Report 251, Ústavu merania SAV, 2012.
- [5] R. Škoviera, K. Valentín, S. Štolc, and I. Bajla. Recognition of Untrustworthy Face Images in ATM Sessions using a Bio-inspired Intelligent Network. In Maria De Marsico, Ana Fred, editor, *Proceedings of the 2nd International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, pages 511–517, Barcelona, Spain, 2013. SciTePress.
- [6] K. Valentín, I. Bajla, and S. Štolc. A performance comparison of HTM and DBN models applied to visual object classification. In J. Maňka, M. Tyšler, V. Witkovský, and I. Frollo, editors, *Proceedings of the 9th Int. Conf. on Measurement*, pages 23–27, Smolenice, Slovakia, May 2013.
- [7] K. Valentín and M. Malý. Network firewall using artificial neural networks. *Computing and Informatics*, 32(6):1312–1327, 2013.