

Софийски университет „Св. Климент Охридски“

Факултет по математика и информатика

Катедра „Компютърна информатика“

**Обобщаващи свойства на слоевете в конволюционните
невронни мрежи**

Антон Недялков Христов

АВТОРЕФЕРАТ

на дисертация за присъждане на образователна и научна степен „доктор“
в професионално направление 4.6 Информатика и компютърни науки
Докторска програма „Информационни системи“

Научни ръководители:
проф. д-р Мария Нишева
доц. д-р Димо Димов

София, 2025

БЛАГОДАРНОСТИ

Айнщайн е казал, че *„въображението е по-важно от знанието“*. Точно тези думи бяха крайъгълният камък, заради който започна това мое пътешествие.

Искам да изкажа моите благодарности към научните ми ръководители проф. д-р Мария Нишева и доц. д-р Димо Димов, които ме подкрепяха през всички тези години. Искрено съм Ви признателен, че уважихте всяко едно мое желание в развитието на научната ми работа и че бяхте с мен по този път.

Без Вашия професионализъм и съвети тази дисертация нямаше да бъде възможна.

ПОСВЕЩАВА СЕ НА

Добри Добрев, който неуморимо заявяваше интереса си към научните ми изследвания. Завладяващите разговори с теб завинаги ще останат в съзнанието ми. Почивай в мир!

Съдържание

Съдържание	4
1. Увод	5
1.1. Актуалност на темата	5
1.2. Цели и задачи на дисертационния труд	6
2. Обзор на предметната област	6
2.1. Теоретични основи	7
2.2. Архитектурни влияния	7
2.3. Влияния на слоевете	7
2.4. Техники за регуляризация	7
2.5. Анализирание на характеристиките	7
2.6. Основни (foundation) модели	7
2.7. Обобщение	8
3. Теоретични основи на обобщаващите свойства в слоевете на конволюционните невронни мрежи	8
3.1. Развитие на конволюционните невронни мрежи и слоевете им	8
3.2. Какво виждат конволюционните невронни мрежи	10
4. Филтрите в конволюционните невронни мрежи като независими детектори на визуални концепти	15
4.1. Избор на архитектура и обучение	15
4.2. Вектор на визуални концепти	15
4.3. Експерименти	16
5. Основен (foundation) модел за откриване на визуални шаблони чрез самоконтролирано обучение	20
5.1. Въведение в новостта на предложения подход	20
5.2. Теоретична постановка на предложения метод	21
5.3. Експерименти	26
6. Заключение	30
6.1. Дискусия	30
6.2. Основни приноси на дисертационния труд	31
6.3. Публикации и доклади, свързани с темата на дисертационния труд	32
6.4. Планове за бъдещо развитие	33
Литература	34
Декларация за оригиналност	36

1. Увод

Изкуствените невронни мрежи (Artificial Neural Networks, ANN) се очертаха като една от най-успешните информационни технологии в съвременното, оказвайки дълбоко влияние върху области като компютърно зрение, обработка на естествен език, роботика и др. Вдъхновени от структурата и функцията на биологичните нервни системи, като човешкия мозък, ANN са клас модели за машинно обучение, предназначени да разпознават модели в данните, да правят прогнози и да решават сложни проблеми чрез обучение върху данни. Тяхната гъвкавост и мащабируемост ги позиционира в челните редици на изследванията и приложенията на изкуствения интелект. Основополагащата концепция за ANNs датира от средата на 20-ти век, с развитието на перцептроните [1] като най-ранните модели на изкуствени неврони. Областта претърпява значителен напредък, от адаптирането на backpropagation алгоритъма [2], до разпространението на архитектури за дълбоко обучение. В основата на ANN лежи нейната слоеста архитектура обикновено състояща се от входни, скрити и изходни слоеве. Всеки слой е съставен от изчислителни възли, наричани неврони, комуникиращи помежду си чрез връзки, съдържаща тегла. Чрез итеративни процеси на обучение ANN коригират тези тегла, за да минимизират грешките и да оптимизират изпълнението на дадена задача.

Конволюционните невронни мрежи (Convolutional Neural Networks, CNN) са вдъхновени от биологичните процеси, тъй като връзката между невроните наподобява организацията на зрителната кора на бозайниците. Те са проектирани да откриват сложни характеристики във визуалните данни, като бележат значителен напредък, главно в обработката на изображения.

От изследванията на Hubel и Wiesel при маймуни и котки [3] е известно, че мозъкът съдържа малки региони от клетки, чувствителни към определени региони от зрителното поле, наречени поле на възприятие. Те са позиционирани така, че да покриват цялото зрително поле. Тези клетки действат като филтри над ограничено входно пространство, откриващи и извличащи единствено локални връзки от съответната област. Идентифицирани са два основни вида клетки: обикновени клетки реагиращи максимално на специфични ръбовидни модели в рамките на своето поле на възприятие; комплексни клетки имащи по-широки полета на възприятие, локално инвариантни спрямо позицията на обектите.

Тъй като зрителната кора на бозайниците е най-мощната съществуваща система за визуална обработка, изглежда естествено нейният модел на действие да бъде приложен в сферата на разпознаването на образи.

1.1. Актуалност на темата

Конволюционните невронни мрежи са се очертали като крайъгълен камък на съвременното машинно обучение, революционизирайки областта на компютърното зрение и обработката на изображения. Тяхната способност за автоматично научаване на йерархични визуални характеристики, главно чрез конволюционни, обединяващи и напълно свързани слоеве, им позволява да постигнат най-съвременна производителност в широк набор от приложения като класификация на изображения, откриване на обекти и семантично сегментиране. Въпреки техния забележителен успех, разбирането на основните механизми, които им позволяват да обобщават добре нови за тях данни, остава активна област на изследване и до днес.

Обобщаващите свойства са главна концепция в машинното обучение, определяща дали и до каква степен даден модел запазва производителността си върху непознати за него примери. Въпреки че CNN често са възхвалявани за тяхното добро представяне в практически задачи, позадълбочено теоретично разбиране за това как и защо те се справят по този начин остава често неуловимо.

Изследването на обобщаващите свойства на слоевете в CNN е както теоретично, така и практическо начинание. На теоретично ниво това включва анализа на различните архитектури и операции в тях, водещи до повишаване на производителността, както интерпретацията и

визуализацията на визуалните концепти, на които реагират отделните слоеве. От практическа страна разбирането на генерализацията позволява използването на вече готовите модели към различно приложение, като може да доведе до по-добър дизайн на моделите, нови и подобрени техники за обучение и по-надеждно внедряване на CNN в сценарии от реалния свят.

Съществуващата литература изследва различни аспекти на обобщаващите свойства на CNN като въздействието на дълбочината, широчината и различните архитектури. Въпреки това, почти никакво внимание не се отделя на изолирането и анализирането на приноса на отделни слоеве към производителността на генерализацията в мрежите. Разбирането на този принос е от съществено значение за оптималното използване на CNN, особено в задачи, където данни липсват или трудно могат да бъдат предоставени.

Изследването на една CNN може да се опрости чрез фокусиране върху три основни типа слоеве, ранни, средни и крайни, всеки от които играе важна и отделна роля в извличането и обработката на характеристики, като техният принос към генерализацията може да варира значително.

1.2. Цели и задачи на дисертационния труд

Основната цел на дисертационния труд е изследване на обобщаващите свойства на слоевете в конволюционните невронни мрежи и разработка на подходи, поддържащи тези свойства, които позволяват решаване с висока производителност на практически задачи, насочени към разпознаването и откриването на обекти в непознати за тях данни.

Основните задачи, свързани с целта на дисертацията, биха могли да се систематизират в:

- Теоретично изследване на развитието на архитектурите на CNN и операции в тях, водещи до повишаване на производителността (секция 3.1);
- Теоретично изследване на интерпретацията и визуализацията на визуалните концепти, на които реагират отделните слоеве в CNN, подкрепено с експерименти (секция 3.2);
- Практическо изследване на обобщаващите свойства на CNN чрез анализиране на приноса на отделни слоеве към производителността върху нови данни (глава 4);
- Разработка на нов подход, доказващ, че всички слоеве от една CNN съдържат в себе си обобщаващи свойства, влияещи върху класификацията на изображения (глава 4);
- Разработка на основен (foundation) модел на CNN за откриване на обекти и шаблони, използващ свойствата на локализация във всички слоеве (глава 5).

Основната насока на научното изследване е чрез доказателства да бъде оборен митът, че единствено крайните слоеве в CNN съдържат обобщаващи свойства и съществени характеристики за решаване на избрана задача.

Хипотезата, направена тук в тази посока, гласи:

Конволюционните невронни мрежи трябва да бъдат разглеждани не като линейни модели, в които всички слоеве участват последователно в решаването на конкретна задача, а като пространствени модели, съдържащи независими и самостоятелни изчислителни единици.

2. Обзор на предметната област

Обобщаващите свойства на CNN са широко проучвани, за да се разбере способността им да се представят добре върху нови за тях данни. Досегашните изследвания в тази област разглеждат различни теоретични основи, архитектурни влияния, специфични за слоевете приноси и усъвършенствани техники за регуляризация и анализиране на характеристиките, които подобряват и интерпретират обобщаващите им свойства. Резултатът от дългогодишното развитие на генерализацията в CNN доведе до появата на основните (foundation) модели с общо предназначение, които са предварително обучени и могат да бъдат адаптирани към широк набор от задачи в компютърното зрение.

2.1. Теоретични основи

Теоретичните изследвания се стремят да обяснят как CNN генерализират добре въпреки големия им брой параметри, които моделите оптимизират по време на обучението си. В тази връзка е предложен метод [4] за оценка на границите на обобщаването, които са теоретични ограничения за това колко добре може да се представи модел върху нови за него данни. Чрез нея е демонстрирано, че оптимизационните методи предпочитат по-прости модели, постигащи с тях по-ефективна генерализация.

2.2. Архитектурни влияния

Архитектурният дизайн на CNN играе важна роля в способността им да обобщават. ResNet [5] въвежда прескачащи връзки, представляващи преки пътища, които заобикалят един или повече слоеве в мрежа. Тези връзки позволяват на моделите да запазят важна информация и да избегнат проблеми като изчезващи градиенти. DenseNet [6] разшава това чрез свързване на всички слоеве директно един към друг. Този подход, известен като повторно използване на характеристиките, позволява на модела да използва предварително научени визуални концепти в цялата си структура. EfficientNet [7] предлага систематичен начин за мащабиране на мрежите, като балансира тяхната дълбочина, широчина и разделителна способност. Хибридни архитектури като ConvNeXt [8] интегрират традиционните CNN с концепции от визуалните трансформатори [9].

2.3. Влияния на слоевете

Различните слоеве в CNN допринасят по различен начин в обучението и извличането на характеристики. Ранните слоеве, които обикновено улавят характеристики на ниско ниво като ръбове и текстури, могат да бъдат лесно прехвърлени към нови задачи и данни. Средните слоеве служат като посредници, научавайки по-абстрактни и специфични за задачата характеристики. Тези слоеве са от решаващо значение в прехвърлянето на знания между различни домейни, както и при обучение на модели, изпълняващи множество задачи едновременно [10]. Крайните слоеве, от друга страна, са специализирани за конкретната задача и са склонни към прекомерно нагаждане, когато не са правилно регуляризирани.

2.4. Техники за регуляризация

Методите за регуляризация са незаменими за подобряване на обобщаването на CNN. Традиционната групов нормализация [11] е от основно значение за стандартизиране на входния информационен поток към всеки слой по време на обучение, стабилизирайки и ускорявайки обучението. Регуляризация на теглата в мрежата [12] помага да се контролира силата на активациите, като по този начин се предотвратява прекомерното нагаждане.

2.5. Анализирани на характеристиките

Една от най-популярните задачи при анализирани на характеристиките в CNN е откриването на близки по вид изображения на база съдържание на сцените и/или обектите в тях. Методите в тази област могат да се групират според мястото на конволюционния слой, чиито CNN карти на характеристиките се акцентират: от последния конволюционен слой [13]; от напълно свързаните слоеве в комбинация с последния конволюционен слой [14]; от първия напълно свързан слой [15]; от конволюционен слой по избор [16].

2.6. Основни (foundation) модели

Основните модели са предварително обучени с голямо количество данни модели с висока степен на генерализация, предназначени да изпълняват повече от една задача с достатъчно висока точност. Тези модели служат като основа за различни приложения, позволявайки дообучение или адаптиране към конкретни задачи с по-малко данни и изчислителни ресурси. Например DINOv2 [17] трансформатор, обучен чрез самоконтролирано обучение, който позволява семантично сегментиране, без да разчита на аотирани данни.

2.7. Обобщение

Обширната литература за обобщаващите свойства на CNN осигурява стабилна основа за анализиране на приноса на отделните слоеве. Въпреки, че е постигнат значителен напредък в разбирането на факторите, влияещи върху генерализацията, остават значителни пропуски в цялостното изолиране и количествено определяне на специфичните за отделните слоеве свойства. Чрез надграждане на предишни изследвания и интегриране на теоретични знания за архитектурите, оптимизацията и анализите, тази работа има за цел да подобри разбирането за обобщаващите свойства на слоевете в конволюционните невронни мрежи и да доведе до проектирането на по-стабилни и интерпретируеми модели.

3. Теоретични основи на обобщаващите свойства в слоевете на конволюционните невронни мрежи

3.1. Развитие на конволюционните невронни мрежи и слоевете им

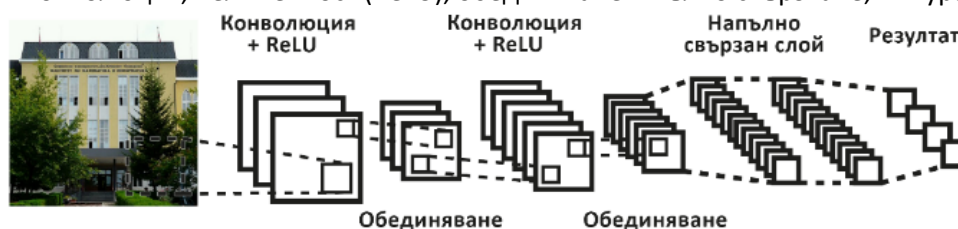
Целта на настоящата глава е да представи как развитието на CNN през годините способства за увеличаване на производителността им до достигане на етап, в който още на ниво архитектура те интегрират в себе си обобщаващи свойства или имат потенциал за това.

3.1.1. Структура и обучение на конволюционните невронни мрежи

Конволюционните невронни мрежи трансформират, слой по слой, пикселните стойности на изображенията в резултат, който ги класифицира към един от класовете, дефинирани заедно с тренировъчните входни данни. Някои слоеве извършват фиксирани математически операции; други съдържат параметри, настройвани така, че резултатите, изчислявани от мрежата, да съответстват на класа, към който принадлежи съответното изображение на входа.

3.1.1.1. Архитектура

Слоеве от неврони, изграждащи гръбнака на една CNN, най-общо реализират 4 основни операции: конволюция, нелинейност (ReLU), обединяване и пълно свързване, Фигура 1.



Фигура 1. Примерна архитектура на CNN.

Конволюционен слой

Главната цел на конволюционния слой е да се извлекат различни характеристики от входното изображение. Той е съставен от определен брой филтри, които обхождат входните за него данни и проверяват за конкретна характеристика. На всяка позиция от обхождането се извършва скалярно умножение на 2 матрици (конкретния филтър и частта от входните данни, препокрити от него), а резултатът е изходът от филтрирането в дадената позиция. Резултатите от всички позиции, през които минава даденият филтър, образуват карти на характеристиките.

Слой нелинейност (ReLU)

Проблемът с изчезващия градиент [18] показва, че при невронни мрежи с активационна функция като сигмоид или хиперболична тангента, при мрежа с N на брой слоеве, градиентът намалява експоненциално с N . Това води до много по-бавно обучение на началните слоеве в сравнение с останалите. Активационната функция „*ректифицирана линейна единица*“ (Rectified Linear Unit, ReLU) [19] е решение на проблема с изчезващия градиент. ReLU се прилага върху всеки пиксел в дадения слой, като в резултат всички отрицателни стойности се заменят с нули. Тъй като конволюцията е линейна операция, чрез ReLU се добавя възможността мрежата да моделира нелинейни функции.

Слой обединяване

Обединяване, приложено върху карта на характеристики, съкращава размерността ѝ, но същевременно запазва (по-)съществената информация. Така характеристиките стават устойчиви срещу шум и изкривяване във входните данни, както и по-малки по размер. Обединяването може да бъде от различен тип: максимално, осреднено, сумарно и т.н. В случай на максимално обединяване резултатът е матрица на максималните стойности от всеки входен регион. При сумарното – резултатът са сумите от всеки входен регион, докато при осредненото – резултатът допълнително се нормализира (целочислено) с размера на входните региони.

Напълно свързан слой

Това е традиционна ANN със софтвакс активационна функция в изходния слой. При него всеки неврон от даден слой е свързан с всеки неврон от предишния слой. Изходът от конволюционните и обединяващите слоеве представя характеристики от високо ниво. Целта на напълно свързания слой е да използва тези характеристики, за да класифицира входното изображение към един от класовете, определени от тренировъчните данни.

3.1.1.2. Обучение

Процесът на обучение на CNN чрез „backpropagation“ метод може да бъде обобщен в:

1. Инициализиране на всички филтри и параметри-(тегла) със случайни стойности.
2. Мрежата приема изображение, прекарва го през слоевете и чрез напълно свързания слой определя принадлежността му както към съответен клас.
3. Изчисляване на общата грешка (loss) в изходния слой, сумарно по всички класове.
4. Обновяване на всички параметри на филтрите използвайки изчислената грешка от предходната стъпка 3, с цел минимизирането ѝ.
5. Повтаряне на стъпки 2-4 с всички изображения от тренировъчното множество, до достигане на желано ниво на общата грешка.

3.1.2. Сравнителен анализ върху развитието на конволюционните невронни мрежи CNN бележат бързото си развитие през 2012 г. Това е годината, когато за първи път те успяват да спечелят състезанието ImageNet [20], което съхранява около 15 милиона анотирани изображения, в общо над 22 000 категории. Главният критерий за успех в това състезание е колко добре една архитектура се справя в класификацията на обекти. Победителите в това състезание и основни CNN, са включени хронологично, като особено внимание е отделено на тези иновации, които допринасят със съществени идеи за развитие на CNN, виж Таблица 1.

Таблица 1. Сравнителен анализ на архитектурите на CNN. Грешка от типа Top-N означава, че алгоритъмът/моделът не успява в рамките на най-добрите си N прогнози да посочи правилната.

Име	Тип	Година	Грешка в ImageNet		Брой параметри	Брой слоеве	Нововъведения
			Top 5	Top 1			
AlexNet [21]	---	2012	15.3%	37.5%	60M	8	Първата CNN, (Top 5) победител в състезанието ILSVRC.
VGG [22]	VGG16	2014	7.2%	24.4%	138M	16	Акцент върху дълбочината на CNN; Редуциране на размера на филтрите;
	VGG19		7.1%	24.4%	144M	19	
ResNet [5]	---	2015	3.6%	19.4%	60M	152	Рекордно ниска грешка 3.6% (Top 5); "Остатъчен" (Residual) блок.
SegNet [23]	---	2016	---	---	14.7M	27	Семантична сегментация на изображения с CNN; Игнорира се напълно свързаният слой.
MobileNetV1 [24]	---	2017	10.5%	29.4%	4.2M	28	CNN за малки устройства с нископроизводителен хардуер; Поканално единична конволюция;
DenseNet [6]	DenseNet-121	2017	6.7%	23.6%	8M	121	Плътна свързаност между слоевете; Свободно протичане на градиента към по-ранните слоеве, смекчавайки проблема с изчезването му;
	DenseNet-169		5.9%	22.1%	14M	169	
	DenseNet-264		5.3%	20.1%	34M	264	

MobileNetV2 [25]	---	2018	---	28.0%	3.4M	53	Подобрение на MobileNetV1; Въвеждане на обърнати остатъчни блокове и линейни гръбнаци;	
EfficientNet [7]	EfficientNet B0	2019	---	6.7%	22.9%	5.3M	241	„Комбинирано мащабиране“, за балансирано мащабиране на размерите на модела; Търсене на мрежова архитектура “NAS” за проектиране на базовия модел B0.
	EfficientNet B2			5.1%	19.9%	9.2M	343	
	EfficientNet B4			3.6%	17.1%	19M	478	
	EfficientNet B7			3.0%	16.7%	66M	817	
ConvNeXt [8]	ConvNeXt-T	2022	---	15.9%	29M	153	Хибриден подход, при който CNN се възползва от иновациите в трансформаторите, без да се изоставят силните страни на конволюцията. Включва в себе си характеристики на основните (foundation) модели.	
	ConvNeXt-S			14.2%	50M	297		
	ConvNeXt-B			13.2%	89M	297		
	ConvNeXt-L			12.5%	198M	297		
	ConvNeXt-XL			12.2%	350M	297		

Тенденциите в развитието на CNN отразяват преминаването от увеличаване на дълбочината и сложността им, започвайки с AlexNet [21] и стигайки ResNet [5] за достигане на точност, позволяваща прилагането им в реални задачи към действия, насочени до увеличаване на ефективността и компактността им чрез MobileNets [24, 25] и EfficientNet [7]. Достигането на високото им ниво на развитие е пряко свързано и с разработки върху анализ и оптимизация на протичащия поток от информация в тях, виж SegNet [23] и DenseNet [6]. Тези комплексни и сложни архитектури в началото са изградени изцяло ръчно, като поради достигане на ограничения в този подход общността е започнала да използва методологията за автоматизираното им проектиране “NAS”. Традиционният, но вече модернизирания CNN модел ConvNext [8], който се адаптира към тенденциите на визуалните трансформатори, през 2022 година успява да ги надскочи по отношение на точност и мащабируемост, постигайки 87.8% точност на ImageNet top-1. Включвайки в себе си характеристиките на основните модели, той става предпочитаната мрежа за задачите в компютърното зрение.

3.2. Какво виждат конволюционните невронни мрежи

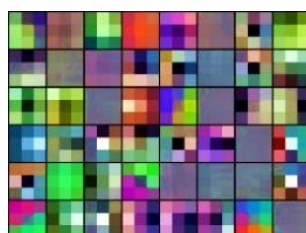
CNN са често критикувани, че не могат да бъдат интерпретирани, затова са наричани „черни кутии“, тъй като е трудно за хората да разберат принципа им на работа. Визуализацията и интерпретацията на слоевете им са от решаващо значение за разбирането на техните свойства за обобщение, тъй като те дават интуитивна представа за това как тези модели обработват данните, извличат определени характеристики и вземат решения. Трансформациите върху изображенията извършвани от слоевете на CNN са нелинейни поради активационните им функции и заради това те трудно биват визуализирани. Докато активационните стойности на първия слой могат директно да бъдат представени в пикселното пространство, връзките между характеристиките в по-дълбоките слоеве на мрежите са много по-сложни.

3.2.1. Методи за визуализация на конволюционните невронни мрежи

В експериментите на изследването е използвана конволюционна невронна мрежа VGG16 [22], предварително обучена върху масива от данни ImageNet [20].

3.2.1.1. Визуализация на филтрите от първия слой на мрежата

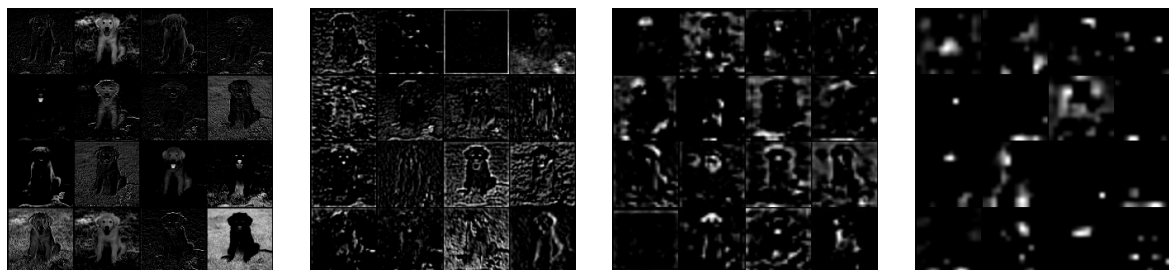
За да се добие представа на каква визуална структура реагира всеки един от филтрите в мрежата е удобно да бъдат визуализирани техните тегла. Това се практикува при филтрите от първи слой, тъй като техните тегла могат да бъдат съпоставени с RGB каналите на изображенията, Фигура 2.



Фигура 2. Визуализация на някои от филтрите от слой 1 на VGG16, обучен върху ImageNet [20].

3.2.1.2. Визуализация на картите на характеристиките

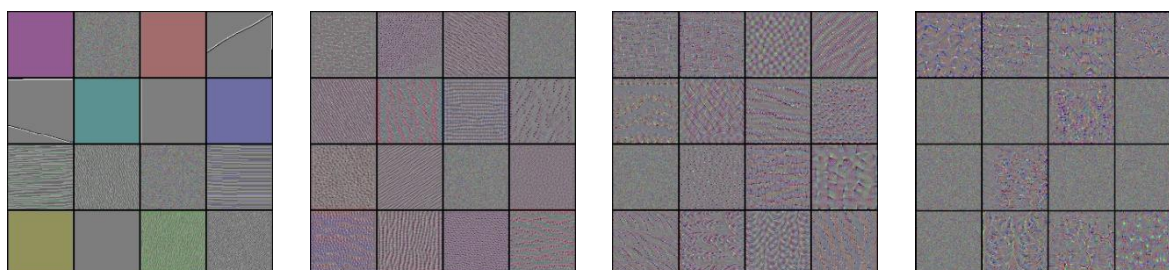
Следващият прост метод за визуализация е разделянето на резултата от прилагането на филтрите в мрежата по канали и възприемането на всеки канал като черно бяло изображение. Всеки канал е една карта на характеристиките, която представлява резултата от прилагане на конкретен филтър. Идеята на това визуализиране за конкретно входно изображение е да се разбере какви характеристики от входа се откриват или запазват от конкретен филтър и слой. На Фигура 3 са показани по няколко карти на характеристиките за избрани слоеве. Наблюдава се очакваното, че картите на характеристиките в близост до входа откриват малки или фини детайли, докато картите, близки до изхода на модела, улавят по-обща характеристики.



Фигура 3. Визуализация на карти на характеристиките на слоеве 1, 5, 9 и 12 от VGG16, обучена върху ImageNet.

3.2.1.3. Максимизиране на активационните стойности

Друг подход към „разбиране“ на работата на CNN е анализът какви входни данни активират максимално дадена карта на характеристиките. Така може да бъде определено на каква визуална структура реагира определен филтър. Този анализ може да се осъществи чрез оптимизационния алгоритъм „gradient ascent“, [26], приложен върху входното пространство от изображения. На всяка итерация в алгоритъма се прилага „gradient ascent“, но не върху параметрите на дадена CNN, а върху стойностите на входното изображение по пиксели, така че да се максимизират резултатите от изчисленията на конкретен филтър.



Фигура 4. Изображения, максимизиращи картите на характеристиките от слоеве 1, 5, 9 и 12 на VGG16, получени след 10к итерации.

Резултатът от този експеримент показва, че някои филтри от слой 1 на мрежата реагират на определени цветове, други – на различни цветови структури, наподобяващи ръбове, трети пък на случаен цветови шум. Но също е видно, че с увеличаване на дълбочината на мрежата (брой слоеве), филтрите започват да реагират на по-сложни структури, Фигура 4.

3.2.1.4. Дълбоки сънища (Deep Dreams)

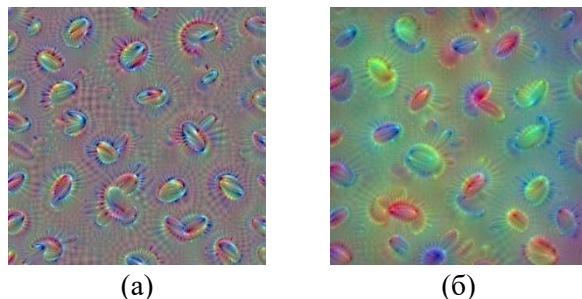
Тази разработка [27] може да се приема като разширение на „Максимизиране на активационните стойности“, като няколко са главните нововъведения, допринасящи за подобряване качеството на визуализациите.

Увеличаване резолюцията на визуализациите

Тъй като входните изображения са с малък размер и визуалните концепти, на които реагират различните филтри, са части от тези изображения, то трудно могат да визуализират техни детайли. Подход, който решава този проблем, е прилагане на алгоритъма „gradient ascent“ в различни мащаби на оптимизираното входно изображение, добавяйки допълнителни детайли към визуализациите на визуалните концепти, както и увеличавайки големината им.

Градиентна нормализация чрез Лапласова пирамида

Резултата от метода за увеличаване резолюцията дава наличие главно на високи честоти, Фигура 5 (а). Метод за подсилване на ниските честоти, предложен отново в [27], е използване на декомпозиция чрез Лапласова пирамида [28], наречен „Градиентна нормализация чрез Лапласова пирамида“. Чрез този подход се добавя заглаждане на градиента на всяка итерация, Фигура 5 (б). Така чрез потискане на високите честоти се постига изравняването им с по-ниските.



Фигура 5. Градиентна нормализация чрез (б) Лапласова пирамида върху (а) входно изображение.

3.2.1.5. Обръщане на характеристики

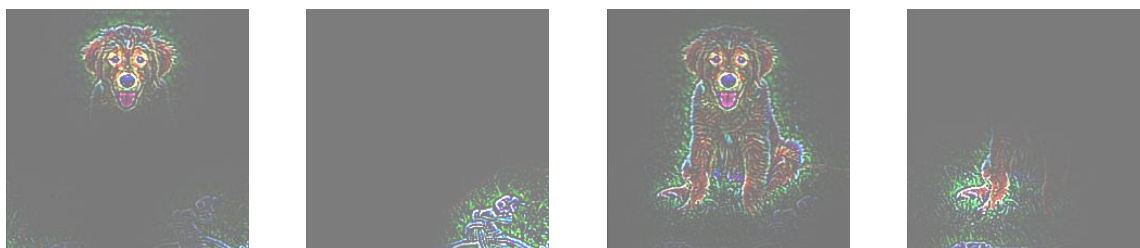
Основната идея на този метод [29] е визуализиране на начина, по който избрана CNN вижда изображенията, чрез извършване на обратна трансформация на характеристиките към изображенията. Първоначално се избира целево изображение, което да бъде анализирано, както и карта/карти на характеристиките от избран слой на мрежата. Втората стъпка е реконструиране на изображение със случайни стойности по такъв начин, че характеристиките на използваната CNN на това изображение да бъдат близки до избраните характеристики на целевото изображение. Реконструкцията се извършва чрез оптимизационен алгоритъм „gradient descent“. От Фигура 6 ясно се вижда как с увеличаване на дълбочината в мрежата слоевете започват да реагират на по-сложни визуални концепти.



Фигура 6. Реконструирани изображения чрез „обръщане на характеристики“, показващи на какво реагират слоевете 1, 5, 9 и 12 във VGG16 (използвани са всички карти на характеристиките от съответния слой).

3.2.1.6. Направляван backpropagation

Задачата на метода „направляван backpropagation“ [30] е да определи кои части на входното изображение на една CNN активират избрана част от мрежата, Фигура 7. Първо се подава изображение на мрежата и се изчисляват активационните стойности (движение напред) на избрана карта на характеристиките. След това се изчислява градиентът между картата и входното изображение (движение назад). Крайното реконструирано изображение, използвайки изчислените градиенти, показва частите от входното изображение, които максимално активират избрания неврон от картата на характеристиките.



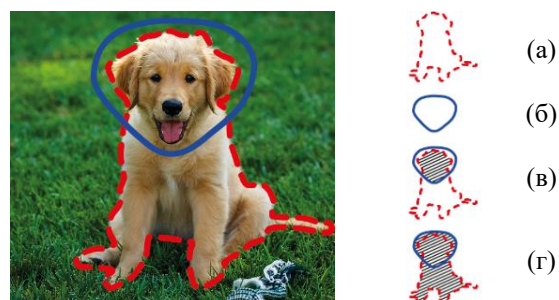
Фигура 7. Визуализация на частите от входно изображение, активиращи произволно избрани карти на характеристиките на слой 13 от мрежата VGG16.

3.2.1.7. Дисекция на мрежа

Подходът „дисекция на мрежа“ [31] количествено определя доколко всяка единица на CNN може да бъде интерпретирана. Той свързва силно активирани области в характеристиките на CNN с „човешки“ понятия (предмети, текстури, цветове, ...). Авторите на [31] създават масив от изображения Broden, Фигура 8, чрез който откриват семантиката на елементите на CNN чрез свързването им с „човешки“ интерпретируеми концепти. Подходът „дисекция на мрежа“ оценява всяка карта на характеристиките от CNN като решение на бинарна задача за сегментиране на всяка визуална концепция в Broden. Резултатът му дава по една активационна маска за всяка карта на характеристиките и всяко изображение. Концептът се намира чрез сравнение на активационните маски с всички концепти от масива. Определя се количествено връзката между активационната маска и концептната маска чрез *IoU* (Intersection over Union) метриката, виж Фигура 9.



Фигура 8. Резултат на няколко интерпретируеми части (от слой 12) от дисекцията на мрежа VGG16.



Фигура 9. Пример за *IoU* на (а) маска на концепт, (б) активационна маска, (в) област на сечение и (г) област на обединение

3.2.1.8. CNN фиксации

CNN фиксации [32] е метод за изчисляване на важните места в изображенията, като не изисква архитектурни промени, допълнително обучение или изчисляване на градиенти. CNN фиксации е подход за визуализация, който използва научените зависимости между характеристиките между два последователни слоя на една CNN, чрез операциите при преминаване напред в мрежата. Тоест, в даден слой, за избрана активационна стойност може да бъде определен наборът от положително свързани активации от предишния слой, които ѝ влияят. С други думи, подходът локализира областите в изображенията, които са отговорни за прогнозата на избраната мрежа, Фигура 10. Изхождайки от неврона, представляващ интерес, методът разчита единствено матрична операция, за да открие най-важните активационни пътища в използваната мрежа. Невроните, съдържащи се в тези активационни пътища, се идентифицират на всеки слой от мрежата, като се определят техните координати, виж Фигура 10 (б). Тези координати се превръщат в маска на отличителните региони чрез прилагане на Гаусов филтър върху тях.



Фигура 10. „CNN фиксации“ (б) върху изображение (а), получавайки маска на отличителните региони (в).

3.2.2. Експерименти

Проведените експерименти могат да бъдат обобщени в три групи: (1) методи, оптимизиращи входно изображение, съдържащо случайни пикселни стойности, които визуализират обобщени визуални форми, показващи на какъв визуален концепт реагира избрана част от мрежата; (2) методи, локализиращи части на входното изображение, които активират максимално избрана

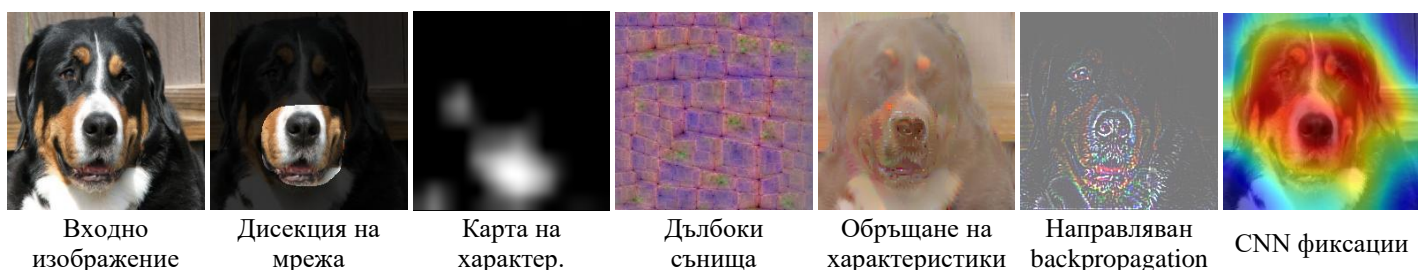
част от мрежата; (3) един метод, отличаващ се от предходните два, който показва на какво точно изображение реагира максимално избрана част от мрежата;

Опитната постановка на експериментите е следната:

- От резултатите на метода „дисекция на мрежа“ са избрани карти на характеристиките с високи *IoU* стойности от различни слоеве. Събрани са семантично свързаните изображения на избраните карти, които ще бъдат използвани като етикет на всяка карта.
- Методите от група 1 са тествани чрез сравняване на визуализациите, които те създават за определената карта на характеристиките и регионите от семантично свързаните ѝ изображения, определени от маските, предоставени от масива Broden.
- Методите от група 2 са тествани чрез анализ на всеки един от тях върху семантично свързаните изображения на избраните карти на характеристиките. Тоест всеки метод определя регионите на входните изображения, активиращи определената карта. Тези региони на входните изображения трябва да съвпадат с регионите на същите, определени от маските, предоставени от масива Broden.

Експеримент 1

Избрана е карта на характеристиките 263 от слой 12 на използваната VGG16 мрежа.

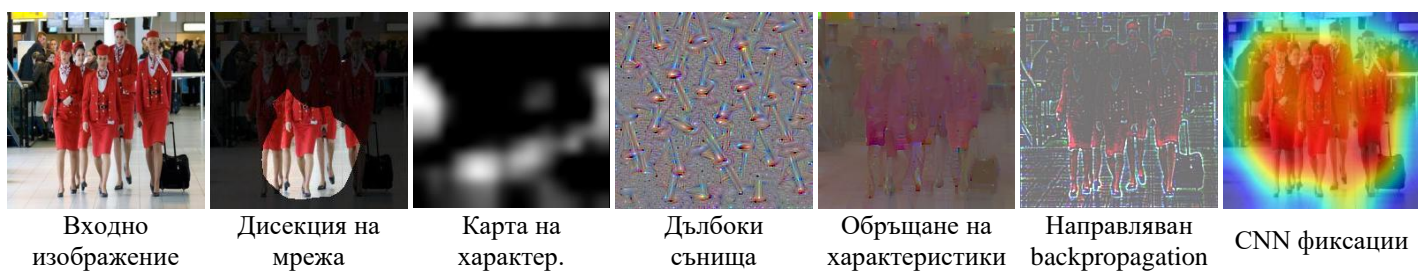


Фигура 11. Сравнени резултати от визуализации на карта на характеристиките 263 от слой 12 на VGG16.

Фигура 11 показва консистентни резултати на методите „дисекция на мрежа“, „визуализация на картите на характеристиките“, „обръщане на характеристики“, „направляван backpropagation“ и до известна степен на „CNN фиксация“. Ясно се вижда разминаването на резултата от метода „дълбоки сънища“.

Експеримент 2

Избрана е карта на характеристиките 37 от слой 12 на използваната VGG16 мрежа.



Фигура 12. Сравнени резултати от визуализации на карта на характеристиките 37 от слой 12 на VGG16.

Фигура 12 показва консистентни резултати на методите „дисекция на мрежа“, „дълбоки сънища“, „обръщане на характеристики“, „направляван backpropagation“ и до известна степен на „CNN фиксация“ и „визуализация на картите на характеристиките“.

Резултатите от експериментите могат да бъдат синтезирани в:

- Наблюдава се значително разминаване между резултата от методите „дълбоки сънища“ и „дисекция на мрежа“. Причината за това може би се корени още в конструирането на оптимизационната задача на gradient ascent, която се стреми към достигане на

максимална стойност на средно аритметичното от активационните стойности на избраната карта на характеристиките.

- Методът „обръщане на характеристики“ се справя в някои експерименти добре, а в други не. Това може би се дължи на обстоятелството, че той е създаден за визуализиране на начина, по който цял слой от CNN мрежа вижда входното изображение, а не избрани отделни карти на характеристиките.
- Методът „направляван backpropagation“ показва най-стабилни резултати.
- Методът „визуализация на картите на характеристиките“, показва добри резултати.
- Методът „CNN фиксации“ се справя задоволително добре, като в общия случай той обхваща по-големи региони от входното изображение в сравнение с другите методи.

4. Филтрите в конволюционните невронни мрежи като независими детектори на визуални концепти

С развитието на конволюционните невронни мрежи е оформен митът, че единствено последните слоеве, предоставят най-съществените характеристики. В тази глава се оборва експериментално това предубеждение чрез демонстрации, показващи, че всички конволюционни слоеве от дадена CNN съдържат в себе си съществена информация, влияеща върху класификацията. В тази връзка всеки филтър от различните конволюционни слоеве се разглежда като независим детектор на визуален концепт и на тази основа въвеждаме ново понятие, наречено „вектор на визуални концепти“ (Vector of Visual Concepts, VVC).

4.1. Избор на архитектура и обучение

За експериментите е използвана мрежата VGG16 [22], избрана поради простата си, но ефективна структура, както и голямата си дълбочина, успяваща да постигне впечатляващи резултати при класификацията на изображения. Тя е обучена и тествана с корпуса от данни CIFAR-10 [33] върху 50 000 тренировъчни и 10 000 тестови изображения от 10 класа.

4.2. Вектор на визуални концепти

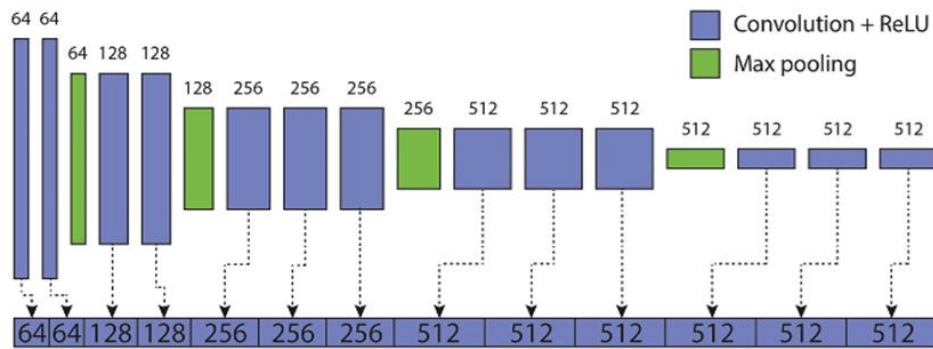
Нека разгледаме CNN не като йерархична структура, спомагаща единствено на научените концепти от високо ниво (съдържащи се в последните слоеве) да правят предположение за обектите, съдържащи се във входните данни, а като структура, съдържаща относително независими един от друг детектори на относително абстрактни визуални структури/концепти. Комбинирането на всички тези детектори би трябвало да дава (достатъчно) изчерпателна информация за съдържанието на входните данни. Тоест нека направим хипотеза, че:

„Текущото състояние на всяка изчислителна единица в CNN влияе директно на класификацията на входните изображения.“

С оглед заключението от изследванията, че филтрите могат да бъдат разглеждани като детектори на визуални структури, можем да конкретизираме нашата хипотеза като:

„Резултатите (характеристиките) от всички конволюционни слоеве в CNN влияят директно върху класификацията на входните изображения.“

Това твърдение е в контраст с разработките през годините, фокусирани върху анализа на характеристиките в CNN. Резюмето на най-важните разработки по темата в последните години показва новостта на предлаганата тук хипотеза. За нейното детайлизиране дефинираме новото понятие „вектор на визуални концепти“ (Vector of Visual Concepts, VVC), като едномерен масив от стойности, всяка от които представя резултата от конкретен филтър, Фигура 13.



Фигура 13. Вектор на VGG16 визуални концепти. Размерът му е равен на броя на картите на характеристиките от всички конволюционни слоеве на мрежата, в този случай = 4224.

VVC дава информация за това, кои детектори на визуални концепти се активират от дадено изображение и колко силно са активирани те. Тоест, VVC за изображения, имащи еднакво или визуално близко съдържание, би трябвало да имат еднакви или близки стойности. Така, търсенето на изображения, съдържащи подобни обекти (еднакви или визуално близки концепти), може да бъде осъществено просто, чрез сравнение на вектори.

Тъй като картите на характеристиките представляват двумерни масиви от стойности, за да бъдат трансформирани във VVC, всяка една от тях трябва да бъде преобразувана в единична стойност, представляваща коефициент, показващ колко силно е активирана картата от текущото входно изображение. Тоест този коефициент показва силата на активация на съответния филтър, който тук наричаме още „детектор на визуален концепт“. Измежду огромния брой възможни трансформации на матрица в единична числова стойност, тук се разглеждат следните три:

T1. Средно аритметичната стойност на всички стойности:

$$f_{mean} = \frac{1}{n} \sum_{i=1}^n a_i \quad (1)$$

T2. Средно аритметичната стойност на всички стойности, по големи от нула:

$$f_{pos} = \frac{1}{n_{pos}} \sum_{i=1}^{n_{pos}} a_i \quad (2)$$

T3. Коефициент на подобие, изчислен по формулата:

$$f_{coef} = \frac{n_{pos}}{n} f_{mean} \quad (3)$$

където: a_i е елемент на i -тата позиция от картата на характеристиките; n е броят на всички елементи от картата на характеристиките; n_{pos} е броят на (строго) положителните елементи в картата на характеристиките.

Всеки конволюционния слой може да бъде разделен на три отделни операции, които заедно с горните три трансформации, дават общо 9 варианта за изчисляване на компонентите на VVC:

- O1. Матрично умножение на входните (за дадения слой) данни с конкретния филтър;
- O2. Добавяне на отклонение към всеки елемент от матричния резултат от предходната O1;
- O3. Прилагане на активационна функция върху резултата от предходната O2.

4.3. Експерименти

Новото понятие VVC, предлагано в тази глава, най-общо показва „силата“, с която дадено изображение активира различните детектори на визуални концепти. Тук е изследвана експериментално възможността за откриване на търсено визуално съдържание чрез стандартна метрика в линейното векторно пространство, асоциирано с VVC.

Експериментираният подход по предлагания тук VVC метод се сравняват със съответната „стандартна“ грешка от класификация: 0.0641 (за етап 1 от експериментите) и 0.2952 (за етап 2). Подходът с най-малка грешка дава, макар и с малко, но по-голяма грешка спрямо т. нар. „стандартна класификация“, която представя разглежданата VGG16 [22], обучена за CIFAR-10

[33]. Причината е в обучението, което преднамерено не е допълнително оптимизирано за новите домейни с входни данни при експериментите. За простота на експериментите тук извършваме класификацията по метода на най-близкия съсед (метода *NN*) с използване на евклидова метрика, стандартно дефинирана за линейното векторно пространство на *VVC*.

Проведени са две фази на експерименти: първата фаза на *CIFAR-10* (т.е. 10 000 тестови изобр. от 10 класа), а втората – на *CIFAR-100* (други 10 000 тестови изобр., но от 100 нови класа).

4.3.1. Етап 1 на експеримента

Най-напред е показано експериментално, че всички карти на характеристиките, получени от конволюционните слоеве на една *CNN*, влияят върху класификацията на входните изображения. За целта са изчислени *VVC* на тренировъчните изображения и на тестовите изображения от масива *CIFAR-10*, използвайки трите формули (1), (2) и (3), приложени върху картите на характеристиките за всеки един от трите етапа на обработка. Така за целите на сравнителния анализ са изчислени множеството на тренировъчните *VVC* и тестовите *VVC* в 9 варианта, като за всеки вариант са изчислени Евклидовите разстояния между конкретен тестови *VVC* и всички тренировъчни *VVC* за дадения вариант. Класът на всяко едно тестово изображение от *CIFAR-10* е определен по метода *NN* в множеството на тренировъчните *VVC* (чиято принадлежност към 10-те класа е известна). Грешката на класификация е определена чрез отношението на погрешните тестови класификации към общия брой на изобр., 10 000 в *CIFAR-10*, Таблица 2.

Таблица 2. Грешката от класификация на тестовите данни от *CIFAR-10* по новия *VVC* подход.

Приложена трансформация		Топ 1 грешка след:		
		(1) Прилагане на филтър	(2) Добавяне на отклонение	(3) Прилагане на активация
1	f_{mean}	0.0682	0.0682	0.0655
2	f_{pos_mean}	0.6107	0.6494	0.0659
3	f_{coef}	0.0653	0.0654	0.0660
Стандартна класификационна грешка:		0.0641	0.0641	0.0641
Минимална разлика:		0.0012	0.0013	0.0014

Резултатите показват, че *VVC* са в състояние да класифицират тестовите изображения с точност, различаваща се с 0.0012 (0.12%) от точността при стандартния подход при работа с *CNN* (само по изходния слой на мрежата). Тъй като използваната *CNN* има напълно свързани слоеве, съдържащи обучаващи се параметри, то в Таблица 3 са дадени резултатите от класификацията чрез *VVC* с добавени коефициентите от теглата на напълно свързаните слоеве в мрежата. Резултатите показват, че добавянето на характеристиките от напълно свързаните слоеве влошава макар и с малко точността на класификация.

Таблица 3. Грешката от класификация на тестовите данни по *VVC* подхода, плюс теглата от напълно свързаните слоеве.

Приложена трансформация		Топ 1 грешка след:		
		(1) Прилагане на филтър	(2) Добавяне на отклонение	(3) Прилагане на активация
1	f_{coef}	0.0653	0.0654	0.0660
2	f_{mean} (напълно свързан слой)	0.0661	0.0661	0.0661
3	f_{pos_mean} (напълно свързан слой)	0.6081	0.6481	0.0659
4	f_{coef} (напълно свързан слой)	0.0663	0.0666	0.0666
Стандартна класификационна грешка:		0.0641	0.0641	0.0641
Минимална разлика:		0.0012	0.0013	0.0018

Разработките през годините използват характеристиките от конкретно избран или последния конволюционен слой на *CNN*. Таблица 4 сравнява резултата от предложението тук *VVC* подход, но приложен само за конкретни конволюционни слоеве, избрани в [13, 16]. Въпреки че предложеният *VVC* метод (ред 1) има най-малка класификационна грешка в сравнение с другите

подходи (редове 2, 3, 4 и 5), идеята на това сравнение е да потвърди, че характеристиките от началните слоеве в CNN могат също да участват във формиране на резултата от класификацията.

Таблица 4. Грешка от класификация по VVC подхода за избрани конволюционни слоеве по трансформация T3.

Приложена трансформация		Топ 1 грешка след:		
		(1) Прилагане на филтър	(2) Добавяне на отклонение	(3) Прилагане на активация
1	f_{coef} (конв. слой 1-13)	0.0653	0.0654	0.0660
2	f_{coef} (конв. слой 13)	0.0659	0.0659	0.0656
3	f_{coef} (конв. слой 12)	0.0661	0.0658	0.0659
4	f_{coef} (конв. слой 11)	0.0692	0.0710	0.0697
5	f_{coef} (конв. слой 10)	0.0852	0.0847	0.0810
Стандартна класификационна грешка:		0.0641	0.0641	0.0641
Минимална разлика:		0.0012	0.0013	0.0015

4.3.2. Етап 2 на експеримента

Етап 2 показва, че детекторите на визуалните концепти, включени във VVC, могат да бъдат използвани върху данни от различен домейн в сравнение с домейна на тренировъчните данни, както и показва влиянието на характеристиките от началните слоеве. От Таблица 4 е видно влиянието на характеристиките от последния конволюционен слой на CNN. Съответно обяснението е, че мрежата се е научила да разпознава много точно концепти от високо ниво, съдържащи се в тренировъчните данни. И в случая точно тези концепти от високо ниво много добре описват важните визуални характеристики на данните.

Какво обаче би станало, ако мрежата бъде използвана върху данни от домейн, различен от тренировъчния? Вероятно концептите от високо ниво от тренировъчния и новия домейн биха били силно (и визуално) отличаващи се, тъй като новите данни ще съдържат различни класове обекти. В този случай на помощ биха дошли концептите от по-ниско ниво в мрежата, тъй като по-простите визуални концепти са по-обобщаващи - изображенията от почти всеки домейн имат визуално близки върхове, ръбове, контури, и т.н..

В този експеримент е използван масива от данни CIFAR-100 [33], който съдържа 60 000 цветни изображения от общо 100 класа, всяко с размер 32×32 пиксела. С него е обучена VGG16, за да бъде оценена минималната класификационна грешка, която тази архитектура може да достигне върху посочените данни.

Таблица 5. Сравняване на класификационната грешка на тестовите данни от CIFAR-100 по стандартния и по VVC подхода, на същата мрежа, но тренирана върху CIFAR-10.

Приложена трансформация		Топ 1 грешка след:		
		(1) Прилагане на филтър	(2) Добавяне на отклонение	(3) Прилагане на активация
1	f_{mean}	0.6221	0.6221	0.6532
2	f_{pos_mean}	0.9127	0.9245	0.6027
3	f_{coef}	0.6638	0.6703	0.6806
Стандартна класификационна грешка:		0.2952	0.2952	0.2952
Минимална разлика:		0.3269	0.3269	0.3075

Първият експеримент е аналогичен на този от Таблица 2. Но тук в Таблица 5 е съпоставена новата „стандартна“ класификационна грешка на VGG16, обучена върху CIFAR-100, и класификационните грешки по VVC метода, но изчислени чрез характеристиките на вече използваната мрежа, обучена върху CIFAR-10. Очевидно, въпреки че CNN е обучавана върху данни от класове обекти, различни от тези на тестовите данни, CNN е успяла да научи достатъчно обобщени характеристики, чрез които се постига сравнително добра класификация.

В Таблица 6 (а), по аналогия с Таблица 4, за „най-добрата“ версия от Таблица 5 (ред 2, колона 3), е направено съпоставяне с VVC, изчислени върху различни групи слоеве на мрежата.

Резултатът потвърждава предположението ни, че последните конволюционни слоеве научават много специфични за тренировъчните данни визуални концепти, силно различаващи се от новите тестови данни. Поради това началните слоеве успяват да класифицират по-точно новите данни в сравнение с крайните. Интересното тук е, че характеристиките от първия и втория конволюционен слой, които реагират на цветовете и ръбовете, успяват да класифицират по-точно новите данни в сравнение с последните слоеве.

Таблица 6. Сравняване на "стандартната" класификационна грешка при CIFAR-100, чрез VVC върху тренираната с CIFAR-10 мрежа: (а) за конкретен конв. слой; и (б) за избрани групи конв. слоеве.

(а) Приложена трансформация	Топ 1 грешка (след активация)	(б) Приложена трансформация	Топ 1 грешка (след активация)
$f_{positive_mean}$ (конв. слоеве 1-13)	0.6027	$f_{positive_mean}$ (конв. слоеве 1-13)	0.6027
$f_{positive_mean}$ (конв. слой 13)	0.8906	$f_{positive_mean}$ (конв. слоеве 1-12)	0.5772
$f_{positive_mean}$ (конв. слой 12)	0.8741	$f_{positive_mean}$ (конв. слоеве 1-11)	0.5699
$f_{positive_mean}$ (конв. слой 11)	0.8325	$f_{positive_mean}$ (конв. слоеве 1-10)	0.5713
$f_{positive_mean}$ (конв. слой 10)	0.7425	$f_{positive_mean}$ (конв. слоеве 1-9)	0.5757
$f_{positive_mean}$ (конв. слой 9)	0.6756	$f_{positive_mean}$ (конв. слоеве 1-8)	0.5795
$f_{positive_mean}$ (конв. слой 8)	0.6708	$f_{positive_mean}$ (конв. слоеве 1-7)	0.5972
$f_{positive_mean}$ (конв. слой 7)	0.6561	$f_{positive_mean}$ (конв. слоеве 1-6)	0.6170
$f_{positive_mean}$ (конв. слой 6)	0.6062	$f_{positive_mean}$ (конв. слоеве 1-5)	0.6405
$f_{positive_mean}$ (конв. слой 5)	0.6263	$f_{positive_mean}$ (конв. слоеве 1-4)	0.6727
$f_{positive_mean}$ (конв. слой 4)	0.6724	$f_{positive_mean}$ (конв. слоеве 1-3)	0.7038
$f_{positive_mean}$ (конв. слой 3)	0.6993	$f_{positive_mean}$ (конв. слоеве 1-2)	0.7614
$f_{positive_mean}$ (конв. слой 2)	0.7636	Стандартна класиф. грешка:	0.2952
$f_{positive_mean}$ (конв. слой 1)	0.8521	Минимална разлика:	0.2747
Стандартна класиф. грешка:	0.2952		
Минимална разлика:	0.3110		

Таблица 6 (б) показва класификационните грешки при използване на VVC върху конволюционните слоеве от 1 до някакъв текущо избран. Резултатите показват, че най-малката грешка се достига при използване на слоеве от 1 до 11, като тя е по-малка от отделните грешки на всеки един от слоевете (сравни с Таблица 6 (а)).

Таблица 7. Сравняване на "стандартната" класификационна грешка при CIFAR-100, чрез VVC върху тренираната с CIFAR-10 мрежа, сравнена с VVC за избрани конволюционни слоеве.

Приложена трансформация	Топ 1 грешка (след активация)
$f_{positive_mean}$ (конв. слоеве 1-13)	0.6027
$f_{positive_mean}$ (конв. слоеве 6-11)	0.6025
$f_{positive_mean}$ (конв. слоеве 5-11)	0.5797
$f_{positive_mean}$ (конв. слоеве 4-11)	0.5671
$f_{positive_mean}$ (конв. слоеве 3-11)	0.5536
$f_{positive_mean}$ (конв. слоеве 2-11)	0.5706
$f_{positive_mean}$ (конв. слоеве 1-11)	0.5699
Стандартна класификационна грешка:	0.2952
Минимална разлика:	0.2584

Резултатите от Таблица 7, базирано на най-малката грешка от Таблица 6 (б) (конволюционни слоеве от 1 до 11) и най-малката грешка от Таблица 6 (а) (конволюционен слой 6), показват, че класификационната грешка започва да намалява чрез добавянето на характеристики от началните конволюционни слоеве. Това още веднъж показва, че върху класификацията на изображенията влияят характеристиките от всички конволюционни слоеве.

5. Основен (foundation) модел за откриване на визуални шаблони чрез самоконтролирано обучение

Намирането на местоположение на шаблон в изображение е основен проблем в много приложения на компютърното зрение като локализиране на обекти, регистриране на изображения, съвпадение на изображения и проследяване на обекти. Наличните понастоящем методи се провалят, когато не са налични достатъчно данни за обучение или когато в изображенията съществуват големи вариации в текстурите, те са от различни модалности или съществуват слаби визуални характеристики, което води до ограничени приложения при задачи от реалния свят.

В тази глава представяме разработения тук основен (foundation) модел за откриване на визуални шаблони (**Self-Supervised Foundation Model for Template Matching, Self-TM**), който представлява нов подход, използващ изцяло самоконтролирано обучение. Идеята зад Self-TM е да бъдат създадени йерархични характеристики, включващи свойства за локализация, обучени върху изображения без никакви анотации. С навлизането по-дълбоко в слоевете на CNN техните филтри започват да реагират на по-сложни структури и техните полета на възприятие се увеличават. Това води до загуба на информация за локализация за разлика от ранните слоеве. Йерархичното разпространение на активациите от последните слоеве обратно към първия слой води до прецизна локализация на шаблоните. Благодарение на възможността си за генерализиране с висока точност върху нови изображения в задачи като извличане на изображения, плътно откриване на шаблони (dense template matching) и разрежено съвпадение на изображения (sparse image matching), нашият Self-TM може да бъде класифициран като основен (foundation) модел.

5.1. Въведение в новостта на предложения подход

Въпреки значителния прогрес в развитието на техниките за откриване на визуални шаблони, разработените до момента решения не акцентират върху достатъчно широк спектър от съществени свойства, важни за практиката, като способност за генерализация, изпълнение в реално време и лесно претрениране, което не изисква анотирани данни, и др.

Подходът, в който Self-TM се използва, може да се причисли към така наречените мрежи сиамски близнаци (Siamese based network) [34], където стандартно един и същ модел (encoder) се използва за извличане на характеристики на изображението на търсения обект, а също и на характеристики на изображението, в което той се търси. След това характеристиките от последните слоеве се корелират чрез обучаемата кръстосана корелация [34] и резултатът след това се обработва от един или няколко декодера (конволюционни невронни мрежи или трансформатори) [5, 8, 9, 35, 36]. Най-често резултатът от декодерите представлява една или няколко „regression map“ [34] и „classification map“ [34].

Базирайки се на този подход, настоящата работа предлага много ефикасна на база брой параметри архитектура, в допълнение с опростен, но в същото време точен корелационен подход. В комбинация с интуитивния метод за обучение Self-TM може лесно да бъде фино настроен (дообучен) върху всякакви типове изображения.

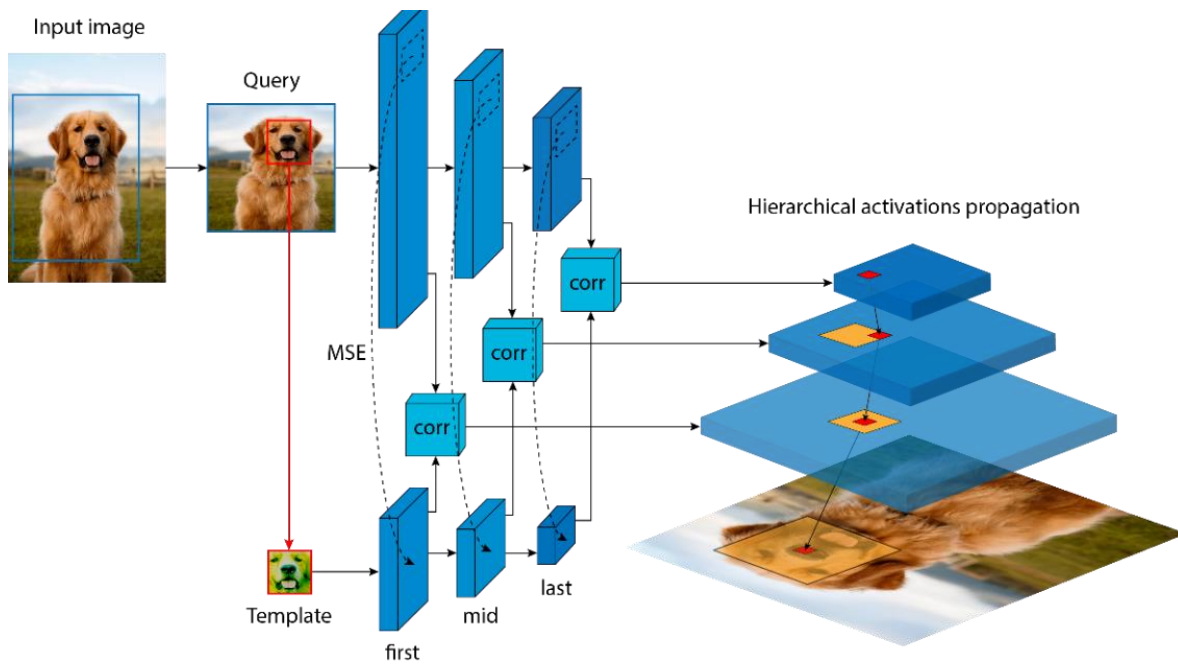
Новостта в предложения Self-TM се изразява в следното:

- Високата степен на генерализация елиминира необходимостта от претрениране на модела с реалните данни, върху които ще бъде използван;
- Ако все пак е необходимо дообучение, необходими са много малко на брой изображения на реалните данни до достигане на желаната точност;
- Специално проектираният модел CNN е обучен върху задачата за локализиране на търсен обект, различавайки се от стандартните подходи за използване на мрежа, обучена върху някаква друга задача, най-често класификационна или по-обща такава;

- Резултатът от използвания корелационен оператор предоставя реална информация за локацията на търсения обект, а не е отделен слой от мрежата [34], който трябва да бъде обучаван и след това резултатът от него да бъде декодиран чрез една или повече ANN. Използването на този подход не е открито в литературата до настоящия момент;
- Йерархично разпространение на активации от последния към първия слой, което води до прецизна локализация, като същевременно изключва необходимостта от използване на допълнителен декодер, който трябва да бъде обучаван. Резултатът е изключително опростена и лека архитектура, която в същото време предоставя висока точност. Използването на този подход не е открито в литературата до настоящия момент;
- Двуетапно самоконтролирано обучение, включващо два типа трансформации на данните: цветови трансформации; цветови и геометрични трансформации;
- Self-TM е ротационно инвариантен. В тази работа Self-TM семейството от модели са обучени в целия интервал от -90 до +90 градуса;

5.2. Теоретична постановка на предложението метод

В този раздел е дискутирана постановката на предложението тук основен (foundation) модел за откриване на визуални шаблони чрез самоконтролирано обучение (Self-TM), виж Фигура 14. Подробно са обяснени техническите подробности като избор на архитектура и включените в нея слоеве, данни и етапи на обучение, както и съществени свойства на йерархичните характеристики, включващи свойства за локализация на шаблони от изображения без никакви анотации. След това са представени експерименталните резултати на Self-TM върху данни с различна модалност и разнообразие от задачи включващи локализация на шаблони, проверка на пачове, откриване на пачове и съвпадение на изображения.



Фигура 14. Илюстрация на Self-TM.

5.2.1. Модел

Избрана е конволюционна невронна мрежа, тъй като концепцията на предложението метод включва използване на йерархичните активации от крайния към първия слой на мрежата, където в колкото по-далечен слой се намира даден неврон, той има по-голямо поле на възприятие (receptive field), тоест обхваща по-голям регион от входните пиксели.

Предложението метод е независим от архитектурата на конволюционната невронна мрежа, като може да бъде прилаган към модели, притежаващи различни качества. Избраната архитектура е базирана на наскоро въведената архитектура ConvNeXt [8], която е подобна на ResNet [5], но е

създадена така, че да се конкурира по производителност със съвременните вижън трансформатори [9, 36]. В допълнение на това, тази архитектура е една от малкото, която се използват ефективно при самоконтролирани обучения.

Self-TM използва блоковете на ConvNeXt [8], но с променен размер на филтрите и стъпката на отместване, виж Таблица 8. Блоковете за намаляне на размера на данните вече използват филтри с размер 3×3 и стъпка на отместване 3, което от гледна точка на полето за възприятие дава на неврон от слой N зрително поле на регион с размер 3×3 от слой $N-1$.

Таблица 8. Детайлно описание на архитектурата на Self-TM. С X_{first} , X_{mid} и X_{last} е означен броят на параметрите съответно в първия, средния и последния слой за всеки размер на модела.

Размер	X_{first}	X_{mid}	X_{last}	Брой параметри
Self-TM Small (малък)	128	256	512	13M
Self-TM Base (среден)	128	384	1024	40M
Self-TM Large (голям)	128	512	2048	130M
Входен размер	Име на слой	Компоненти в слой		Изходен размер
$3 \times 189 \times 189$	Намаляне на размерност	[Conv2D 3×3 , $[X_{first}]$, stride 3] Norm		$X_{first} \times 63 \times 63$
$X_{first} \times 63 \times 63$	ConvNeXt блок	[Conv2D 7×7 , $[X_{first}]$, stride 1, pad 3] Norm, Linear $[X_{first}, 512]$ GELU, Linear $[512, X_{first}]$] $\times 3$		$X_{first} \times 63 \times 63$
$X_{first} \times 63 \times 63$	Нормализация	Norm		$X_{first} \times 63 \times 63$
$X_{first} \times 63 \times 63$	Намаляне на размерност	[Norm Conv2D 3×3 , $[X_{mid}]$ stride 3]		$X_{mid} \times 21 \times 21$
$X_{mid} \times 21 \times 21$	ConvNeXt блок	[Conv2D 7×7 , $[X_{mid}]$, stride 1, pad 3] Norm, Linear $[X_{mid}, 1024]$ GELU, Linear $[1024, X_{mid}]$] $\times 9$		$X_{mid} \times 21 \times 21$
$X_{mid} \times 21 \times 21$	Нормализация	Norm		$X_{mid} \times 21 \times 21$
$X_{mid} \times 21 \times 21$	Намаляне на размерност	[Norm Conv2D 3×3 , $[X_{last}]$, stride 3]		$X_{last} \times 7 \times 7$
$X_{last} \times 7 \times 7$	ConvNeXt блок	[Conv2D 7×7 , $[X_{last}]$, stride 1, pad 3] Norm, Linear $[X_{last}, 2048]$ GELU, Linear $[2048, X_{last}]$] $\times 3$		$X_{last} \times 7 \times 7$
$X_{last} \times 7 \times 7$	Нормализация	Norm		$X_{last} \times 7 \times 7$

Размерът на Self-TM Small е значително компресиран – съдържа 13 милиона тренировъчни параметри, което е значително по-малко от стандартните архитектури, Таблица 9. Това е породено от дизайна на мрежата, изцяло фокусиран върху такъв тип задачи.

Таблица 9. Сравнение на размера на Self-TM със стандартните използвани архитектури.

Архитектура	Брой параметри
Self-TM Small	13M
DeiT-S [36], ViT-S [35], ConvNeXt-T [8]	22-29M
Self-TM Base	40M
ConvNeXt-S [8]	50M
EffNet-B7 [7], DeiT-B [36], ViT-B [35], ConvNeXt-B [8]	66-89M
EffNetV2-L [37]	120M
Self-TM Large	130M
ConvNeXt-L [8]	198M
ViT-L [35], ConvNeXt-XL [8]	304-350M

5.2.2. Данни

За обучение на Self-TM моделите е използван корпусът от данни ImageNet-1K Train [20], тъй като той се е утвърдил като стандарт за обучение на мрежи, обработващи изображения, както и поради високото разнообразие от класове (1000 класа), обхващащи достатъчен брой различни визуални концепти за достигане на висока генерализация.

5.2.3. Обучение

Използван е често подход за самоконтролирано обучение базиран на инвариантност [38], където идеята е да се научат подобни характеристики за съвместими изображения и различни характеристики за несъвместими изображения.

Обучението е осъществено на два последователни етапа, различаващи се по типа приложени трансформации на входните изображения: цветови трансформации; цветови и геометрични трансформации. Това е необходимо, тъй като геометричните трансформации добавят завишена вариация в данните, която моделът първоначално не може да преодолее. За да бъде преодоляна тази пречка, първоначално моделът бива обучен единствено чрез данни с приложени цветови трансформации. След това, моделът продължава обучението си, като вече към цветовите трансформации се добавят и геометрични. Приложено е една и също обучение върху всички модели Self-TM, започвайки с 15 епохи с цветови трансформации и продължавайки с 30 епохи с цветови и геометрични трансформации, като се използва един и същ набор от данни. За обучението са използвани 4 броя графични карти NVIDIA A5000 24GB, като една епоха отнема около: 5.1 часа за Self-TM Small, 5.7 часа за Self-TM Base и 6.5 часа за Self-TM Large.

Стъпките на обучение са следните:

1. От база от данни с изображения без анотации е взето входно изображение I , върху което е приложено случайно изрязване R , и след това оразмеряване S до 189×189 пиксела, получавайки „query“ изображение, $S(R(I)) = Q$, виж Фигура 14;
2. Върху Q се извършва друго случайно изрязване R и след това се прилага случайна цветова и/или геометрична трансформация A , получавайки „template“ изображение, $A(R(Q)) = T$, виж Фигура 14. Тази стъпка може да произведе един или множество различни шаблони. При Self-TM модела са използвани само два броя шаблони;
3. Запазва се реалната позиция, gt_p , на получения „template“ върху „query“ (координатите на центъра на червения правоъгълник върху „query“ от Фигура 14);
4. „Query“ и двата броя „template“ се подават като вход на Self-TM мрежата, $f_\theta: (Q, T) \rightarrow (y, y')$, като получените карти на характеристиките от всички слоеве се запазват, $y = f_\theta(Q), y' = f_\theta(T)$, $(y, y') = \{(y_n, y'_n) | n = 1, \dots, N\}$, където N е броят на слоевете в мрежата. В Self-TM броят на слоевете е 3, означени с: „first“, „middle“, „last“;
5. Върху всеки две съответстващи си карти на характеристиките, започвайки от най-дълбоката, се прилага корелационен оператор, $CORR(y, y')$, като в получения резултат, се търси позицията на максималната стойност, $pred_p_n = softmax(CORR(y_n, y'_n))$;
6. Върху всеки две съответстващи си карти на характеристиките се изчислява средна квадратна грешка, $MSE_n = MSE(y_n, y'_n) = \frac{1}{|D|} \sum_D (R(y)_n - y'_n)^2$, $D = def(y_n) \cap def(y'_n)$, където D е областта на сумиране, т.е. сечението на двете дефиниционни области, на картата на характеристиките на „template“ y'_n и региона от картата на характеристиките на „query“ $R(y)_n$, $(y, y') = \{(y_n, y'_n) | n = 1, \dots, N\}$. Умножението е скалярно, т.е. поелементно в D ;
7. Извършване на оптимизация на параметрите (gradient descent): чрез минимизиране на грешките, получени от MSE_N, \dots, MSE_1 , и отместване на позициите на $pred_p_N, \dots, pred_p_1$ спрямо реалните позиции на шаблона gt_p_N, \dots, gt_p_1 .

5.2.4. Трансформации на изображенията

Цветови трансформации

Приложените цветови трансформации са базирани на подхода в [39].

За получаване на „query“ изображението се прилагат последователно:

- Изрязване с коефициент за мащабиране: от 0.1 до 0.9
- Оразмеряване до 189×189 пиксела
- Нормализация: $mean = [0.485, 0.456, 0.406]$, $std = [0.228, 0.224, 0.225]$

За получаване на двата „template“ върху „query“ се прилагат последователно:

- Промяна на цветовете с вероятност* на прилагане 80%: $brightness = 0.4$, $contrast = 0.4$, $saturation = 0.2$, $hue = 0.1$
- Обезцветяване с вероятност* на прилагане 20%
- За „template“ 1:
 - Гаусово размазване: радиус от 0.1 до 0.2
- За „template“ 2:
 - Гаусово размазване с вероятност* на прилагане 10%: радиус от 0.1 до 2.0
 - Обръщане на всички пиксели над даден праг с вероятност* на прилагане 20%:
 $thresh = 128$
- Нормализация: същата като при „query“
- Изрязване (съотношение от 0.2 до 5.0) с мащабиране (от 0.14 до 0.85)

* Случайно прилагане на операцията с избрана вероятност, по закона за равномерно разпределение.

Таблица 10. Визуализация на случайни цветови трансформации върху случайно изрязани изобр. от ImageNet-1K.

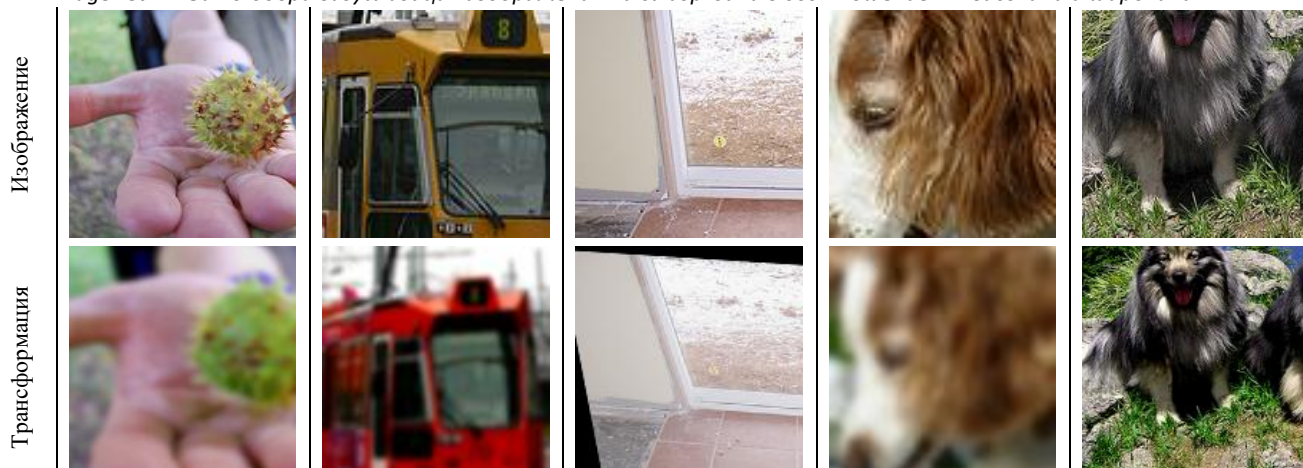


Геометрична трансформация

Геометричните трансформации (подравнени по центъра на получения квадрат за изрязване) се прилагат единствено върху „template“ изображенията:

- Изчисляване на квадрат за изрязване с коефициент за мащабиране: от 0.14 до 0.45
- С вероятност 50% се прилага случайно избрана геометрична трансформация:
 - Перспективна трансформация: $distortion_scale = 0.5$
 - Завъртане: с градуси от -90 до $+90$
 - Мащабиране: с коефициент от -0.7 до 1.3

Таблица 11. Визуализация на случайни цветове и случайни геометрични трансформации върху случайни изобр. от ImageNet-1K. За по-добра визуализация изображенията са изрязани в съотношение 1:1 височина и широчина.



5.2.5. Йерархично разпространение на активации

Едно от главните нововъведения в предложението Self-TM е методът за йерархичното разпространение на активации от последния към първия слой, $pred_{p_N}, \dots, pred_{p_1}$ (в текущата архитектура: $pred_{p_{last}}, pred_{p_{mid}}, pred_{p_{first}}$), което води до прецизна локализация, като същевременно изключва необходимостта от използване на допълнителен декодер, който трябва да бъде обучаван.

Локализирането на „template“ T в избрано изображение Q се осъществява чрез откриване на максималната стойност в резултата от корелационния оператор между всеки два съответни слоеве, започвайки от най-крайния, $pred_{p_{N,\dots,1}} = \text{softmax}(\text{CORR}(y_{N,\dots,1}, y'_{N,\dots,1}))$. След откриване на позицията на максималната стойност $pred_{p_N}$ в последния слой N , тази информация се разпространява в предходния слой $N-1$ като позиция $pred_{p_{N-1}}$ на предходната максимална стойност, която се търси в ограничен регион, явяващ се полето за възприятие RF на максималната стойност от N , $pred_{p_{N-1}} = \text{softmax}(RF_{pred_{p_N}}(\text{CORR}(y_{N-1}, y'_{N-1})))$. По този начин всеки следващ слой прецизира предполагаемата позиция на търсения „template“ T .

За откриване на позицията на максималната стойност се използва „softmax“ функция, която представлява диференцируема „argmax“ функция, даваща възможност градиентът свободно да протича от края до началото на мрежата по време на обучението:

$$\text{softmax}(z)_i = \frac{e^{-\beta z_i}}{\sum_{j=1}^K e^{-\beta z_j}} \text{ for } i = 1, \dots, K, \text{ and } z = (z_1, \dots, z_K) \in \mathbb{R}^K. \quad (4)$$

5.2.6. Корелация

Друго нововъведение в предложението подход е интегрирането на статичен корелационен оператор в мрежата, чийто резултат дава информация за локацията на търсения обект. Този оператор не е отделен слой като в [34], който трябва допълнително да бъде обучаван и след това резултатът от него да бъде декодиран чрез една или повече невронни мрежи.

Корелационният оператор, използван тук, е нормализирана кръстосана корелация и представлява мярка за подобие на база пикселни стойности.

$$\text{CORR}(x, y) = \frac{\sum_{x', y'} T(x', y') I(x+x', y+y')}{\sqrt{\sum_{x', y'} T(x', y')^2} \sqrt{\sum_{x', y'} I(x+x', y+y')^2}} \quad (5)$$

където $(x', y') \in \text{def}(Q) \cap \text{def}(T)$, т.е. сумирането е само върху сечението на дефиниционните области за входното изображение I и шаблона T , позициониран в точката (x, y) на I .

5.2.7. Изчисляване на грешката при обучение
 Два компонента, MSE и $CORR$, се използват за изчисляването на грешката \mathcal{L} :

$$\mathcal{L}_{MSE} = \frac{1}{N} \sum_{n=1}^N \sqrt{MSE(y_n, y'_n)}, y_n \in \mathbb{R}, y'_n \in \mathbb{R} \quad (6)$$

$$\mathcal{L}_{CORR} = \frac{1}{N} \sum_{n=1}^N \sqrt{(pred_p_n - gt_p_n)^2}, pred_p_n \in \mathbb{Z}^2, gt_p_n \in \mathbb{R}^2 \quad (7)$$

$$\mathcal{L} = \frac{\mathcal{L}_{MSE} + \mathcal{L}_{CORR}}{2}, \quad (8)$$

Тук \mathcal{L}_{MSE} е средно аритметичната сума на резултатите от MSE за всеки слой, а \mathcal{L}_{CORR} е средно аритметичната сума на евклидовите разстояния между позициите на откритите максимални стойности $pred_p$ и реалните позиции на шаблоните gt_p , за всеки слой.

5.2.8. Хиперпараметри и оптимизация
 Използван е оптимизатор AdamW [12] със скорост на обучение 0.001 и регуляризация на теглата 10^{-6} , използвана в изчисляването на грешката:

$$\mathcal{L} = \mathcal{L} + \underline{weight_decay} * \underline{L2\ norm\ of\ the\ weights}. \quad (9)$$

5.2.9. Използване на обученния модел
 Получавайки картите на характеристиките от всички слоеве на мрежата за входно изображение $y = f_{\theta}(I_Q)$ и избран шаблон $y' = f_{\theta}(I_T)$, се прилага корелационен оператор $CORR(y, y')$ върху всеки две съответстващи си характеристики. Започвайки с $CORR$ резултата на най-дълбоките слоеве, чрез метода за йерархично разпространение на активации се стига до максималната стойност на активация в първия слой. Позицията на тази активация в картата на характеристиките се явява позицията на шаблона I_T във входното изображение I_Q . Но, тъй като полето на възприятие в използваните блокове за намаляване на размера на данните за слой N е с размер 3×3 от слой $N - 1$, то тази получена позиция е приложима към входното изображение I_Q с 3 пъти намален размер.

5.3. Експерименти

В този раздел са описани експерименталните резултати и демонстрация на предложението тук метод Self-TM върху данни както с еднаква, така и с различна на обучението модалност, както и разнообразие от задачи.

5.3.1. ImageNet-1K Test

Тестовата част на използвания за обучение масив от данни ImageNet [20] е използвана за сравнение, на база точността на локализиране на шаблони, на Self-TM семейството от модели. Резултатът от тестовете е представен в Таблица 12, където за трите размера Self-TM модели, е изчислено отместването в пиксели на трите слоя ($D_{L_{last}}, D_{L_{mid}}, D_{L_{first}}$) на $pred_p_N, \dots, pred_p_1$ спрямо реалните позиции на шаблона gt_p_N, \dots, gt_p_1 .

Таблица 12. Резултати върху ImageNet-1K Test на Self-TM, обучени чрез цветови или цветови и геометрични трансформации. Изчислените отмествания $D_{L_{last}}, D_{L_{mid}}, D_{L_{first}}$ са върху последен, среден и първи слой.

Модел	Трансформации	$D_{L_{first}}$ пиксела	$D_{L_{mid}}$ пиксела	$D_{L_{last}}$ пиксела
Self-TM Small	цветови	0.579	0.176	0.156
Self-TM Base	цветови	0.577	0.173	0.156
Self-TM Large	цветови	0.572	0.171	0.153
Self-TM Small	цветови и геометрични	2.214	0.767	0.409
Self-TM Base	цветови и геометрични	1.752	0.602	0.338
Self-TM Large	цветови и геометрични	1.331	0.452	0.273

Стъпките на тест са подобни на тези при обучението и са следните:

1. От база от данни без анотации (ImageNet-1K Test) се взема входно изображение, от което се получава „template“, чрез който се изчисляват позициите на максималните стойности от резултата на корелационния оператор върху всеки две съответстващи си карти на характеристиките (виж стъпки на обучение от 1 до 5 в т. 5.2.3);
2. За всеки две съответстващи си карти на характеристиките, тоест за всеки слой, се изчислява отместването в пиксели на позициите $D_{L_{last}}, D_{L_{mid}}, D_{L_{first}}$ на $pred_{p_N}, \dots, pred_{p_1}$ спрямо реалните позиции на шаблона $gt_{p_N}, \dots, gt_{p_1}$, изчислени чрез Евклидово разстояние:

$$D_{L_n} = |gt_{p_n} - pred_{p_n}| = \frac{1}{M} \sum_{i=1}^M \sqrt{(gt_{p_n}(i) - pred_{p_n}(i))^2}, \quad (10)$$

където M е броят на изображенията в корпуса от данни, $gt_{p_n}(i)$ и $pred_{p_n}(i)$ са двумерните вектори на всяко изображение, $i = 1, \dots, M$, а n е текущо избран слой от мрежата $\{D_{L_n} | n = 1, \dots, N\}$. В случая $N = 3$ и за прегледност вместо $n = 1, 2, 3$ сме индексирали чрез $first, mid, last$. D_{L_N} или $D_{L_{last}}$ е средното Евклидово разстояние за най-вътрешния слой.

Таблица 13. Визуализация на резултата от открити шаблони в случайни изображения от ImageNet-1K Test. Върху шаблоните са приложени случайни трансформации. Легендата на цветовете е: жълт – позицията на шаблона; червен – изчислената позиция, от най-дълбокия слой; зелен – от средния слой; син – от първия слой.


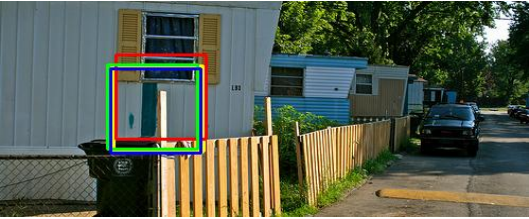
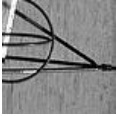

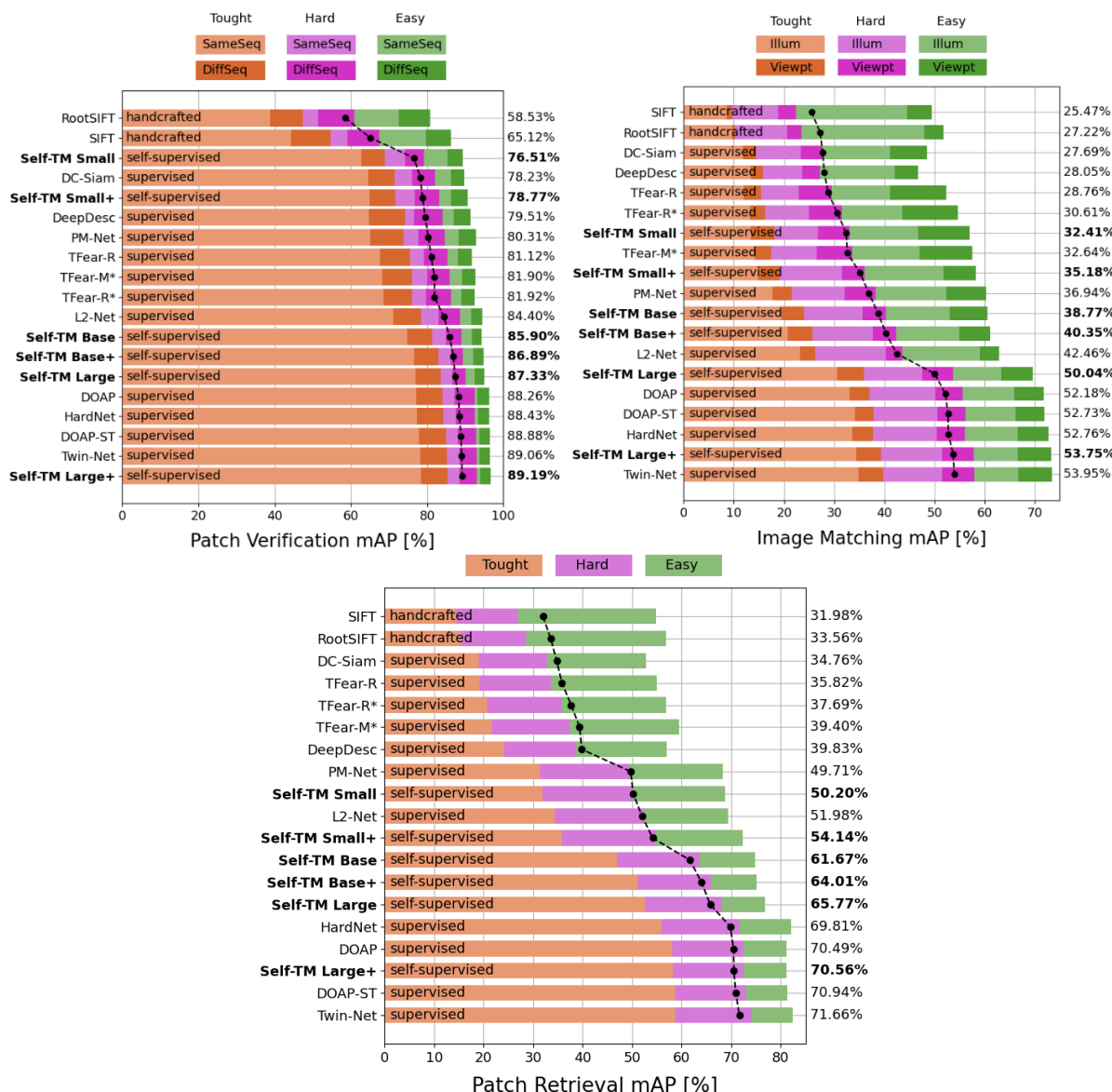
Шаблон за търсене	Резултат
	
	

Таблица 12 показва, че моделите обучени и тествани единствено чрез цветови трансформации, въпреки разликата в размера си успяват да достигнат еднаква точност на локализиране: $D_{L_{first}} = 0.57, D_{L_{mid}} = 0.17, D_{L_{last}} = 0.15$. Това показва, че Self-TM Small (13M) има достатъчно параметри, способни да достигнат до успешно обучение, конкурирайки се с Self-TM Large (130M). Това не е валидно обаче, когато към цветовете трансформации се добавят и геометрични, където при Self-TM Small наблюдаваме почти два пъти по-голямо отместване ($D_{L_{first}} = 2.214, D_{L_{mid}} = 0.767, D_{L_{last}} = 0.409$) в сравнение с Self-TM Large ($D_{L_{first}} = 1.331, D_{L_{mid}} = 0.452, D_{L_{last}} = 0.273$). Това е така, тъй като геометричните трансформации добавят висока вариация в данните за обучение, за което са необходими повече параметри.

5.3.2. HPatches

HPatches [40] е масив от данни за откриване на съвпадения между различни пачове, чрез който е оценено свойството на генерализация на Self-TM. Тези данни имат различна модалност от използваните за обучение, съдържащи сцени с по 6 изображения във всяка с различна осветеност и геометрични трансформации. Съвпаденията на пачовете са открити чрез ключови точки между първото от шестте изображения и останалите. За оценка е използвана метриката mAP върху три различни задачи: проверка на пачове (patch verification), съвпадение на изображения (image matching) и откриване на пачове (patch retrieval).



Фигура 15. Визуално представяне на резултатите върху HPatches (стойностите, без тези за Self-TM, са взети от Twin-Net[41]). Методите са групирани в групите: „handcrafted“, които са ръчно създадени от авторите си; „supervised“, които са използвали анотирани данни за обучението си; „self-supervised“, които не са използвали никакви анотации. С плюс (+) са обозначени Self-TM моделите, дообучени върху корпуса от данни HPatches.

Сравнение на Self-TM с различни методи е показано на Фигура 15. И при трите задачи цветът на маркерите показва количеството „геометричен шум“, лесен (easy), труден (hard) и тежък (tough), съдържащ се в тестовите изображения. Показаните проценти точност представляват осреднените стойности на избраните подзадачи. В експеримента са използвани Self-TM Small, Self-TM Base, Self-TM Large, обучени върху Imagenet1K Train [20] с цветови и геометрични трансформации, като с плюс (+) са обозначени моделите, които са дообучени върху HPatches. Резултатите доказват високата степен на генерализация на моделите, тъй като резултатът на Self-TM Large и при трите задачи е впечатляващ – конкурентен на предишните резултати на модели, обучени с контролирано обучение, доближаващ се до най-добрите. Резултатите на модела Self-TM Large+, дообучен върху HPatches, дори превъзхожда предишните резултати при задачата за проверка на пачове, а при съвпадение на изображения отстъпва първото място с разлика едва 0.2% mAP.

5.3.3. Съвпадение на изображения

Съвпадението на изображения чрез разреждени методи е подходяща задача за проверка на високата степен на генерализация на Self-TM. При нея съвпадението се осъществява чрез

откриване на съответстващи си ключови точки, разпръснати на произволни позиции в набор от изображения, което води до намалени изчислителни разходи и висока устойчивост, Фигура 16. Използвани са широко разпространените данни за проверка точността на съвпадение, имащи силно различаващи се модалности от данните за обучение на Self-TM: MegaDepth [42] и ScanNet [43]. Оценката за точността на съвпадение между всяка двойка изображения е извършена чрез намиране на позицията на камерата чрез изчисляване на съществената матрица с RANSAC [44], разложена след това на ротация и трансляция. Накрая е отчетена ъгловата грешка между изчислената и реалната ротация за всички двойки изображения, представена чрез AUC (area under the ROC Curve) при прагове 5°, 10° и 20°, виж [42, 43].



Фигура 16. Сравнение на OmniGlue [45] (вляво) и OmniGlue + Self-TM Base (вдясно) при откриване на съвпадения между ключови точки в изображение с различна модалност от данните за обучение и на двата модела. За целта на визуализацията, съвпаденията с висока точност не са визуализирани, за да бъдат видими грешките.

Тъй като Self-TM сам по себе си представлява модел CNN, той не може да бъде приложен самостоятелно за решаване на задачата. Поради това е включен във вече разработен метод OmniGlue [45], проектиран и фокусиран към висока степен на генерализация, използващ основния (foundation) модел DINOv2 [17]. Self-TM се включва в метода OmniGlue [45], замествайки DINOv2 [17], без да бъде извършвано каквото и да било негово дообучение. По този начин се извършва пряко сравнение на качеството и генерализацията между характеристиките на Self-TM и известния DINOv2 [17].

Таблица 14. Сравнение между разработения нов метод OmniGlue+ Self-TM спрямо OmniGlue [45] и различни негови конкурентни подходи (изискващи обучение). Резултатите (без тези на OmniGlue+Self-TM) са взети от [45].

Тип подход	Метод	MegaDepth-1500	ScanNet
		AUC@5° / 10° / 20°	AUC@5° / 10° / 20°
Дескриптори и ръчни правила	SIFT [46] + MNN	25.8 / 41.5 / 54.2	1.7 / 4.8 / 10.3
	SuperPoint [47] + MNN	31.7 / 46.8 / 60.1	7.7 / 17.8 / 30.6
Разредени (sparse) методи	SuperGlue [48]	42.2 / 61.2 / 76.0	10.4 / 22.9 / 37.2
	LightGlue [49]	47.6 / 64.8 / 77.9	15.1 / 32.6 / 50.3
	OmniGlue [45]	47.4 / 65.0 / 77.8	14.0 / 28.9 / 44.3
	OmniGlue + Self-TM Small	48.2 / 64.7 / 73.8	15.8 / 29.4 / 43.4
	Разлика (в %) спрямо OmniGlue	+1.8 / -0.4 / -5.1	+13.0 / +1.8 / -2.0
	OmniGlue + Self-TM Base	56.7 / 69.4 / 78.1	22.0 / 34.8 / 47.0
Разлика (в %) спрямо OmniGlue	+19.6 / +6.7 / +0.3	+57.1 / +20.5 / +6.2	
	OmniGlue + Self-TM Large	59.8 / 70.6 / 78.4	26.6 / 37.7 / 48.4
	Разлика (в %) спрямо OmniGlue	+26.2 / +8.7 / +0.8	+90.1 / +30.3 / +9.2

Таблица 14 представя сравнение между новия метод OmniGlue+Self-TM и стандартния OmniGlue [45] и различни негови конкурентни подходи (изискващи обучение), следвайки извършените експерименти в [45]. Тези експерименти включват и по-стандартни традиционни подходи (SIFT

[46] и SuperPoint [47]), използвани все още в задачи, където не е възможно обучение. При тях за откриването на съвпадение между ключовите точки е използван подходът за взаимен най-близък съсед. Таблица 14 показва значително преимущество на OmniGlue+Self-TM Large/Base спрямо стандартния OmniGlue [45] както върху MegaDepth [42], така и върху ScanNet [43]. OmniGlue+Self-TM Small повишава точността при AUC @5°, а изостава при AUC @20°.

Освен акцентирание на точността, в сравнението на DINOv2 [17] и Self-TM трябва да се отчете и размерът на архитектурите им, който пряко влияе на времето за обучение и скоростта на изпълнение (inference). Използваната архитектура DINOv2 е ViT-14-base [9], имаща 87M параметъра, в сравнение с Self-TM Base, имащ 40M, а Self-TM Large – 130M параметъра. Отчитайки резултатите от Таблица 14 и размера на използваните архитектури, може да се стигне до заключение, че Self-TM съдържа в себе си по-висока степен на генерализация в сравнение с DINOv2 (ViT-14-base), като в същото време предоставя и по-оптимизирана от гледна точка на брой параметри и скорост мрежа, Таблица 15.

Таблица 15. Сравнение на скоростта и размера на DINOv2 и Self-TM. Експериментът е извършен с входни изображения с различна резолюция върху процесор Intel Xeon Gold 5222 (3.80GHz) и не е използвана графична карта.

Модел	Брой параметри	Скорост на изпълнение (inference speed)		
		364 × 238 пиксела	742 × 490 пиксела	1498 × 994 пиксела
Self-TM (Small)	13M	212 ms	659 ms	2 481 ms
Self-TM (Base)	40M	244 ms	914 ms	3 432 ms
DINOv2 (ViT-14-base)	87M	445 ms	3 065 ms	38 709 ms
Self-TM (Large)	130M	377 ms	1 268 ms	4 706 ms

6. Заключение

Дисертационният труд представя задълбочено изследване на обобщаващите свойства на слоевете в конволюционните невронни мрежи чрез комбиниране на теоретични анализи с практически експерименти при различни задачи. Работата е фокусирана към справяне с въпроси относно архитектурните компоненти, динамиката на обучението, стратегиите за визуализация и възможностите за извличане на характеристики, които позволяват на CNN да постигнат стабилно и ефективно обобщение на нови данни. Резултатът от проведеното проучване и анализ са разработените новаторски методи, съчетаващи генерализиращи свойства, за решаване с висока производителност на практически задачи, насочени към разпознаването и откриването на обекти в непознати данни.

6.1. Дискусия

Дисертацията започва с подчертаване на важноста на разбирането на генерализацията в CNN, особено предвид тяхното доминиране в приложенията, използващи машинно обучение и компютърно зрение. Тя предоставя солидна обосновка за изучаване на свойствата на обобщаване на CNN слоевете и предлага нови методи за техния анализ.

Извършен е подробен и основополагащ анализ, изследващ теоретичните основи на архитектурите на CNN. Проследява се развитието им през годините с акцент върху нововъведенията им, обхващайки основни архитектури като [5, 6, 7, 21, 22], както и по-нови модели [8]. Сравнителен анализ разкрива специфичните роли на дълбочината на слоевете и размера на филтрите, демонстрирайки как структурните вариации влияят върху способността на моделите да извличат йерархични характеристики и да обобщават.

Важна част на текущото изследване е разбирането на вътрешното функциониране на CNN чрез визуализация на техните характеристики, чрез подходи предоставящи цялостна картина на това как филтрите действат като независими детектори на визуални концепти. Експериментите доказват, че началните слоеве улавят визуални характеристики на ниско ниво, като ръбове и

текстури, докато по-дълбоките слоеве извличат абстрактна семантична информация. Резултатите подчертават как йерархичното представяне позволява на CNN да извършват генерализация, допринасяйки за техния успех в различни задачи.

Изследването въвежда нов метод за анализиране на филтрите на CNN като независими детектори на визуални концепти, предлагайки новото понятие „вектор на визуални концепти“ (VVC). Силно противоречащо на разработките през годините, в които се твърди, че единствено последните слоеве на мрежите научават важни характеристики, VVC демонстрира, с разлика $\sim 0.12\%$ от стандартната класификация, че всички конволюционни слоеве влияят върху класификацията. Експериментите, проведени с масиви от данни с различна модалност, потвърждават, че тези всички филтри играят решаваща роля за постигането на генерализация чрез създаване на значими характеристики.

Нов принос в дисертацията е въвеждането на семейство от основни модели (Self-TM) за прецизно локализиране на шаблони, извличане на изображения, откриване на плътно съвпадение на шаблони и разрежено съвпадение на изображения с помощта на техники за самоконтролирано обучение. Предложеният метод използва йерархично разпространение на активациите, итеративна оптимизация и методи за трансформация на входните изображения, за да подобри производителността на генерализирането. Обучен единствено върху ImageNet-1K, без наличие на никакви анотации, Self-TM предоставя висока степен на генерализация, справяйки се успешно върху данни с различна модалност, като HPatches [40], MegaDepth [42] и ScanNet [43]. При задачата за разрежено съвпадение на изображения Self-TM Base значително превъзхожда DINOv2 [17] с $+19.6\%/+6.7\%/+0.3\%$ (AUC@5°/10°/20°) върху MegaDepth-1500 [42] и с $+57.1\%/+20.5\%/+6.2\%$ (AUC@5°/10°/20°) върху ScanNet-1500 [43]. Върху HPatches [40] резултатите му се доближават до точността на методите, използващи подходи за обучение с учител. При необходимост от повишаване на прецизността, моделът може лесно и бързо да бъде дообучен върху всякакви типове изображения, превъзхождайки предишните резултати, обучени с учител, при задачата проверка на пачове (Patch verification) върху HPatches. Способността на Self-TM да работи без никакви анотации подчертава нарастващото значение на самоконтролираното обучение за намаляване на зависимостта от големи анотирани набори от данни, като по този начин се позволява висока мащабируемост и адаптивност.

6.2. Основни приноси на дисертационния труд

Въз основа на гореописаните изследвания и резултати приносът на този дисертационен труд може да бъде формулиран в следните припокриващите се категории:

Научни приноси

- Хронологичен и сравнителен анализ на еволюцията на CNN
 - Аналитичен обзор на развитието на CNN: Анализ на еволюцията на CNN, от ранни до напреднали архитектури, осигуряващ систематично разбиране на напредъка, който е оформил съвременните CNN.
 - Идентифициране на ключови елементи към дизайна на CNN: Изследването подчертава структурните и функционални компоненти, които допринасят за подобрена производителност, като остатъчни връзки, поканални единични конволюции и ефективни стратегии за мащабиране.
- Теоретични основи за визуализация на CNN и нов подход за интерпретиране:
 - Разбиране на характеристиките: Анализирани и използвани са усъвършенствани методи за визуализация и анализиране на CNN, предлагащи теоретична основа за извличането на йерархични характеристики.
 - Въвеждане на ново понятие „вектор на визуални концепти“ (VCC): На база направената хипотеза, че *„характеристиките от всички конволюционни слоеве в CNN влияят директно върху класификацията на входните*

изображения“, VCC дава информация за това, кои детектори на визуални концепти се активират от дадено изображение и колко силно са активирани те.

Научно-приложни приноси

- Филтрите като независими детектори на визуални концепти:
 - Разработен е нов метод за използване на филтрите във вече обучена CNN като независими детектори на визуални концепти, чрез VCC, подобрявайки разбирането на семантичните и пространствени представяния в характеристиките на мрежите, предоставяйки висока степен на генерализация.
- Основен (foundation) модел за откриване на шаблони:
 - Семейство Self-TM модели: Проектираното семейство от ефективни Self-TM модели са обучени върху задачата за локализиране на търсен обект;
 - Йерархично разпространение на активации: Едно от главните нововъведения в Self-TM е метод за йерархичното разпространение на активации от последния към първия слой, което води до прецизна локализация, същевременно изключвайки необходимостта от използване на допълнителен обучаем декодер.
 - Нов метод за самоконтролирано обучение: Разработен е нов метод за самоконтролирано обучение на два етапа, без никакви анотации при откриване на визуални шаблони, предоставящ висока степен на генерализация.

Приложни

- Обучено семейство Self-TM модели: Self-TM предоставя обучени модели с различни размери, за да отговорят на нуждите на различни приложения. По-малките модели се грижат за приложения в реално време или с ниска мощност, докато по-големите се справят отлично със задачи, изискващи висока точност. Тази мащабируемост гарантира, че предложените модели на CNN са адаптивни към разнообразните изисквания за внедряването в реалния свят и изследователски задачи.

6.3. Публикации и доклади, свързани с темата на дисертационния труд
Резултатите от анализите и извършените научни изследвания в работата по текущата дисертация са апробирани в:

Публикации

- Антон Христов, Мария Нишева, Димо Димов, Въведение в конволюционните невронни мрежи. Автоматика и Информатика, Съюз по автоматика и информатика „Джон Атанасов“ (САИ), ISSN: 0861-7562, бр. 1 (2018), стр. 27-38
- Anton Hristov, Maria Nisheva, Dimo Dimov, Filters in Convolutional Neural Networks as Independent Detectors of Visual Concepts. ACM International Conference Proceeding Series, 2019, pp. 110-117. SJR: 0.200 (2019), <https://dl.acm.org/doi/10.1145/3345252.3345294>
- Anton Hristov, Dimo Dimov, Maria Nisheva, Self-Supervised Foundation Model for Template Matching, Big Data and Cognitive Computing, ISSN: 2504-2289, Special issue “Perception and Detection of Intelligent Vision”, 2025, 9(2):38. JIF: 3.7 (2023) / Q1 (Computer Science, Theory and Methods), <https://doi.org/10.3390/bdcc9020038>

Презентации на научни конференции

- Anton Hristov, Maria Nisheva, Dimo Dimov, Filters in Convolutional Neural Networks as Independent Detectors of Visual Concepts. CompSysTech annual conference 2019 (Ruse, Bulgaria, 21-22 June 2019), <http://www.compsystech.org/docs/CST19-Programme.pdf>

Доклади

- Антон Христов, Какво виждат конволюционните невронни мрежи. Методи за визуализация на вътрешните им слоеве. Анализ на консистентността на резултатите от визуализацията. Пролетна научна сесия на ФМИ, 2020

Награди за най-добра статия (best paper award)

- Anton Hristov, Maria Nisheva, Dimo Dimov, Filters in Convolutional Neural Networks as Independent Detectors of Visual Concepts. ACM International Conference Proceeding Series, 2019, pp. 110-117. SJR: 0.200 (2019), <http://www.compsystech.org/docs/CP2019.pdf>

Забелязани цитирания

- На „Anton Hristov, Maria Nisheva, Dimo Dimov, Filters in Convolutional Neural Networks as Independent Detectors of Visual Concepts. ACM International Conference Proceeding Series, 2019, pp. 110-117. SJR: 0.200 (2019), <https://dl.acm.org/doi/10.1145/3345252.3345294>“:
 - Xiao, L., B. Wu, Y. Hu, J. Liu. A Hierarchical Features-Based Model for Freight Train Defect Inspection. IEEE Sensors Journal, ISSN 1530-437X, Vol. 20, Issue 5 (March 2020), pp. 2671-2678.
 - Mübarek Mazhar, Gökalp Çınar. Mechanical Object Parts Detection using Deep Learning based YOLO Models. International Research in Engineering Sciences III, 2022, <https://doi.org/10.5281/zenodo.7496767>
 - Amaje, Getu Genene. Sweet Potato Leaf Disease Detection and Classification Using Convolutional Neural Network. Doctoral dissertation. 2022.

6.4. Планове за бъдещо развитие

Съществуват няколко направления, в които настоящите изследвания могат да бъдат подобрени и развити, като разработване на основен (foundation) модел за семантично сегментиране използващ йерархичното разпространение на активациите, както и внедряването на база от знания в задачите за обработка и разпознаване на изображения.

Създаването на основен (foundation) модел за семантична сегментация ще се фокусира върху подобряването на генерализацията, ефективността и мащабирането, като същевременно ще се справя с предизвикателства като устойчивост, интерпретиране и адаптивност между различните домейни. Ключова област на изследването би била генерализирането и адаптирането на домейни, което да позволява на модела да се прехвърля ефективно в разнообразни задачи и данни с минимално дообучение, като се използва самоконтролирано и мултимодално обучение. Йерархичното разпространение на активациите, което позволява на моделите да прецизират прогнозите си чрез последователна слой по слой обработка на характеристиките, е друг обещаващ път за подобряване на ефективността на семантичното сегментиране.

Базите от знания могат значително да подобрят свойствата за обобщаване на CNN слоевете чрез предоставяне на структурирана и контекстуална информация, която допълва обучението на моделите. Чрез включването на знания, специфични за домейните, биха се обогатили характеристиките въвеждайки контекстуална информация, позволявайки на CNN да научават връзки и йерархии, които подобряват производителността при нови данни. Ограниченията, базирани на знания, могат да действат като регуляризация пряко намалявайки прекомерното нагаждане (overfitting). Те също могат да участват в увеличаването на синтетичните данни, разширявайки разнообразието от данни за обучение в съответствие с концепти от реалния свят.

В ера на нарастващо внедряване на AI в приложения от реалния свят, генерализирането остава едно от най-критичните предизвикателства в машинното обучение. Тази работа не само

задълбочава разбирането за CNN, но също така предлага практически насоки за проектиране на модели, които са ефективни, интерпретируеми и адаптивни. Анализите, описани тук, проправят пътя за по-нататъшни иновации в системите за дълбоко обучение, като гарантират, че те отговарят на изискванията на все по-сложни и динамични задачи. Тъй като системите за машинно обучение продължават да се развиват, констатациите от тази дисертация ще допринесат за разработването на надеждни и мащабируеми решения за предизвикателствата на утрешния ден.

Литература

- [1] Rosenblatt, Frank. "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review* 65.6 (1958): 386.
- [2] LeCun, Yann, et al. "Backpropagation applied to handwritten zip code recognition." *Neural computation* 1.4 (1989): 541-551.
- [3] Hubel, David H., and Torsten N. Wiesel. "Receptive fields and functional architecture of monkey striate cortex." *The Journal of physiology* 195.1 (1968): 215-243.
- [4] Arora, Sanjeev, et al. "Stronger generalization bounds for deep nets via a compression approach." *International conference on machine learning*. PMLR, 2018.
- [5] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
- [6] Huang, Gao, et al. "Densely connected convolutional networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
- [7] Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." *International conference on machine learning*. PMLR, 2019.
- [8] Liu, Zhuang, et al. "A convnet for the 2020s." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022.
- [9] Dosovitskiy, Alexey. "An image is worth 16x16 words: Transformers for image recognition at scale." *arXiv preprint arXiv:2010.11929* (2020).
- [10] Long, Mingsheng, et al. "Learning transferable features with deep adaptation networks." *International conference on machine learning*. PMLR, 2015.
- [11] Ioffe, Sergey. "Batch renormalization: Towards reducing minibatch dependence in batch-normalized models." *Advances in neural information processing systems* 30 (2017).
- [12] Loshchilov, Ilya and Frank Hutter. "Decoupled weight decay regularization." *arXiv preprint arXiv:1711.05101* (2017).
- [13] Radenovic, Filip, Giorgos Tolias, and Ondrej Chum. "Deep shape matching." *Proceedings of the European conference on computer vision (ECCV)*. 2018.
- [14] Babenko, Artem, et al. "Neural codes for image retrieval." *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I* 13. Springer International Publishing, 2014.
- [15] Sharif Razavian, Ali, et al. "CNN features off-the-shelf: an astounding baseline for recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2014.
- [16] Kim, Been, et al. "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)." *International conference on machine learning*. PMLR, 2018.
- [17] Oquab, Maxime, et al. "Dinov2: Learning robust visual features without supervision." *arXiv preprint arXiv:2304.07193* (2023).
- [18] Hochreiter, Sepp. "Untersuchungen zu dynamischen neuronalen Netzen." *Diploma, Technische Universität München* 91.1 (1991): 31.
- [19] Nair, Vinod, and Geoffrey E. Hinton. "Rectified linear units improve restricted boltzmann machines." *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010.
- [20] Deng, Jia, et al. "Imagenet: A large-scale hierarchical image database." *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2009.

- [21] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems* 25 (2012).
- [22] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *ICLR*, 2015
- [23] Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla. "SegNet: A deep convolutional encoder-decoder architecture for image segmentation." *arXiv preprint arXiv:1511.00561* 5 (2015).
- [24] Howard, Andrew G. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." *arXiv preprint arXiv:1704.04861* (2017).
- [25] Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
- [26] Erhan, Dumitru, et al. "Visualizing higher-layer features of a deep network." *University of Montreal* 1341.3 (2009): 1.
- [27] Mordvintsev, Alexander, Christopher Olah, and Mike Tyka. "Inceptionism: Going deeper into neural networks." *Google research blog* 20.14 (2015): 5.
- [28] [PMIP] Adelson, Edward H., et al. "Pyramid methods in image processing." *RCA engineer* 29.6 (1984): 33-41.
- [29] Mahendran, Aravindh, and Andrea Vedaldi. "Understanding deep image representations by inverting them." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.
- [30] Springenberg, Jost Tobias, et al. "Striving for simplicity: The all convolutional net." *arXiv preprint arXiv:1412.6806* (2014).
- [31] Bau, David, et al. "Network dissection: Quantifying interpretability of deep visual representations." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
- [32] Mopuri, Konda Reddy, Utsav Garg, and R. Venkatesh Babu. "Cnn fixations: an unraveling approach to visualize the discriminative image regions." *IEEE Transactions on Image Processing* 28.5 (2018): 2116-2125.
- [33] Krizhevsky, Alex, and Geoffrey Hinton. "Learning multiple layers of features from tiny images." (2009): 7.
- [34] Hu, Weiming, et al. "Siammask: A framework for fast online object tracking and segmentation." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.3 (2023): 3072-3089.
- [35] Steiner, Andreas, et al. "How to train your vit? data, augmentation, and regularization in vision transformers." *arXiv preprint arXiv:2106.10270* (2021).
- [36] Touvron, Hugo, et al. "Training data-efficient image transformers & distillation through attention." *International conference on machine learning*. PMLR, 2021.
- [37] Tan, Mingxing, and Quoc Le. "Efficientnetv2: Smaller models and faster training." *International conference on machine learning*. PMLR, 2021.
- Proceedings, Part V* 16. Springer International Publishing, 2020.
- [38] Assran, Mahmoud, et al. "Self-supervised learning from images with a joint-embedding predictive architecture." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.
- [39] Bardes, Adrien, Jean Ponce, and Yann LeCun. "Vicregl: Self-supervised learning of local visual features." *Advances in Neural Information Processing Systems* 35 (2022): 8799-8810.
- [40] Balntas, Vassileios, et al. "HPatches: A benchmark and evaluation of handcrafted and learned local descriptors." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
- [41] Irshad, Aman, et al. "Twin-net descriptor: Twin negative mining with quad loss for patch-based matching." *IEEE Access* 7 (2019): 136062-136072.
- [42] Li, Zhengqi, and Noah Snavely. "Megadepth: Learning single-view depth prediction from internet photos." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
- [43] Dai, Angela, et al. "Scannet: Richly-annotated 3d reconstructions of indoor scenes." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

- [44] Fischler, Martin A., and Robert C. Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography." *Communications of the ACM* 24.6 (1981): 381-395.
- [45] Jiang, Hanwen, et al. "OmniGlue: Generalizable Feature Matching with Foundation Model Guidance." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024.
- [46] Lowe, David G. "Distinctive image features from scale-invariant keypoints." *International journal of computer vision* 60 (2004): 91-110.
- [47] DeTone, Daniel, Tomasz Malisiewicz, and Andrew Rabinovich. "Superpoint: Self-supervised interest point detection and description." *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018.
- [48] Sarlin, Paul-Edouard, et al. "Superglue: Learning feature matching with graph neural networks." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020.
- [49] Lindenberger, Philipp, Paul-Edouard Sarlin, and Marc Pollefeys. "Lightglue: Local feature matching at light speed." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023.

Декларация за оригиналност

Декларирам, че настоящият дисертационен труд за присъждане на образователната и научна степен „доктор“ е оригинална разработка и съдържа оригинални резултати, получени при проведени от мен научни изследвания (с подкрепата и/или съдействието на научните ми ръководители).

Декларирам, че резултатите, които са получени, описани и/или публикувани от други учени, са надлежно и подробно цитирани в библиографията.

Декларирам, че настоящата дисертация не е представяна в друг университет, институт или друго висше училище за придобиване на научна степен.

Дата:

Декларатор:

/Антон Недялков Христов/