

**Politechnika Wrocławskaw  
Wydział Informatyki i Telekomunikacji**

---

Kierunek: **Informatyczne systemy automatyki (ISA)**

Specjalność: **Inteligentne systemy przemysłu 4.0 (IPS)**

**PRACA DYPLOMOWA  
MAGISTERSKA**

**Wybrane aspekty modelowania  
neuronowego z uwzględnieniem  
zakłóceń kontradyktoryjnych**

**Adversarial disturbances  
in neural network modeling**

Kamil Pawlik

Opiekun pracy  
**dr hab. inż. Paweł Wachel**

Słowa kluczowe: klasyfikacja, uczenie maszynowe, zakłócenia kontradyktoryjne



## **Streszczenie**

Praca skupia się na tematyce zakłóceń w modelowaniu neuronowym, ze szczególnym uwzględnieniem zakłóceń o charakterze stochastycznym oraz kontradyktryjnym. W pracy przedstawiono aktualny stan wiedzy w zakresie wpływu ww. zakłóceń na proces klasyfikacji obrazów, zwłaszcza na zależność dokładności klasyfikacji od energii zakłóceń. Skupiono się na badaniach szumów białych oraz szumu Cauchy'ego, który do tej pory nie został szczegółowo opisany w literaturze. Zbadano wpływ ataków kontradyktryjnych: *FGSM* [1] oraz *DeepFool* [2], a także zapostulowano, przebadano i porównano autorski algorytm *LESGSM*. Zaproponowano i przebadano szereg strategii regularyzacji szumem i ich efektywności w obniżaniu ogólnej skuteczności ataków kontradyktryjnych.

## **Abstract**

The study focuses on the topic of disturbances in neural modeling, with particular emphasis on stochastic and adversarial disturbances. The paper presents the current state of knowledge regarding the impact of the above-mentioned distortions on the image classification process, especially on the dependence of the classification accuracy on the interference energy. The focus was on research on white noise and Cauchy noise, which has not been described in detail in the literature so far. The impact of adversarial attacks: *FGSM* [1] and *DeepFool* [2] was examined, and the proprietary *LESGSM* algorithm was proposed, evaluated and compared. A number of noise regularization strategies and their effectiveness in reducing the overall effectiveness of adversarial attacks have been suggested and studied.

# Spis treści

<b>1. Wstęp . . . . .</b>	<b>6</b>
1.1. Cel pracy . . . . .	7
1.2. Zakres pracy . . . . .	7
<b>2. Problematyka zakłóceń w modelowaniu neuronowym . . . . .</b>	<b>8</b>
2.1. Źródła zakłóceń . . . . .	8
2.2. Zakłócenia stochastyczne . . . . .	10
2.2.1. Wpływ szumu na klasyfikację obrazów . . . . .	10
2.2.2. Obszary wymagające dalszych badań . . . . .	14
2.3. Zakłócenia kontradydktoryjne . . . . .	14
2.3.1. Atak <i>FGSM</i> . . . . .	14
2.3.2. Atak <i>DeepFool</i> . . . . .	16
2.4. Propozycja zakłócenia kontradydktoryjnego indukowanego stochastycznie . . . . .	19
<b>3. Metodologia badań . . . . .</b>	<b>21</b>
3.1. Środowisko uruchomieniowe . . . . .	21
3.2. Biblioteki programistyczne . . . . .	21
3.3. Zbiory danych i modele neuronowe . . . . .	22
3.3.1. Model podstawowy <i>DNN</i> . . . . .	22
3.3.2. Model konwolucyjny <i>CNN</i> . . . . .	23
3.3.3. Modele rozszerzone . . . . .	24
3.4. Zastosowane metryki . . . . .	24
3.4.1. Dokładność klasyfikacji . . . . .	25
3.4.2. Energia perturbacji . . . . .	25
3.4.3. Macierz błędu . . . . .	26
3.5. Standardowy zestaw badawczy . . . . .	26

<b>4. Badania symulacyjne . . . . .</b>	<b>29</b>
4.1. Wpływ zakłóceń stochastycznych na proces klasyfikacji obrazów . . . . .	29
4.1.1. Wpływ szumu białego o rozkładzie jednostajnym . . . . .	30
4.1.2. Wpływ szumu białego o rozkładzie Normalnym . . . . .	31
4.1.3. Wpływ szumu Cauchy'ego . . . . .	32
4.1.4. Konkluzja wyników analizy symulacyjnej . . . . .	33
4.2. Wpływ zakłóceń kontradyktoryjnych na proces klasyfikacji obrazów . . . . .	33
4.2.1. Wpływ zakłóceń ataku <i>FGSM</i> . . . . .	34
4.2.2. Wpływ zakłóceń ataku <i>LESGSM</i> . . . . .	35
4.2.3. Wpływ zakłóceń ataku <i>DeepFool</i> . . . . .	36
4.2.4. Konkluzja wyników analizy symulacyjnej . . . . .	37
4.3. Ocena skuteczności strategii regularyzacji szumem . . . . .	37
4.3.1. Porównanie strategii regularyzacji szumem dla ataku <i>FGSM</i> . . . . .	38
4.3.2. Porównanie strategii regularyzacji szumem dla ataku <i>LESGSM</i> . . . . .	40
4.3.3. Porównanie strategii regularyzacji szumem dla ataku <i>DeepFool</i> . . . . .	42
4.3.4. Konkluzja wyników analizy symulacyjnej . . . . .	44
4.4. Ocena efektywności algorytmu <i>LESGSM</i> . . . . .	44
<b>5. Podsumowanie . . . . .</b>	<b>48</b>
<b>Literatura . . . . .</b>	<b>50</b>

# Rozdział 1

## Wstęp

Współczesne systemy sztucznej inteligencji osiągają zadowalające wyniki w zadaniach klasyfikacyjnych, często przewyższając ludzkie możliwości w tej dziedzinie. Zdolność ta jest niezwykle istotna w perspektywie nieustannego wzrostu ilości informacji, jaką generujemy każdego dnia. Już dziś algorytmy te wspomagają nas w wielu obszarach życia. Przykładami zastosowań są: analiza ryzyka biznesowego, diagnostyka medyczna, przetwarzanie języka naturalnego, rekommendacje produktów czy rozpoznawanie mowy i obrazów.

W istotnym stopniu przyczynił się ku temu intensywny rozwój technik uczenia maszynowego, a zwłaszcza sieci neuronowych. Te wielowarstwowe systemy do przetwarzania informacji są bardzo wszechstronne, gdyż w teorii spełniają założenia **uniwersalnego aproksymatora** [3]. Oznacza to, że mogą one przybliżyć każdą obliczalną funkcję matematyczną z dowolnym przybliżeniem. W praktyce jednak, dla skomplikowanych problemów klasyfikacyjnych okupione jest to wzrostem liczby neuronów **McCullocha-Pittsa** [4] w warstwach ukrytych sieci, a co za tym idzie wzrostem zapotrzebowania na moc obliczeniową i czasu przeznaczanego na proces uczenia.

Niemniej jednak, mimo nieustanego postępu w tej dyscyplinie, systemy klasyfikacji podatne są na różnego rodzaju zakłócenia. Niezauważalne dla ludzkiego oka zaburzenia w danych wejściowych mogą zmniejszyć zaufanie modelu i powodować błędy klasyfikacji. Sklonność ta występuje zarówno w wypadku zakłóceń o charakterze losowym (błędy, dryfy, szумy) jak i w wypadku intencjonalnych zniekształceń, których celem jest zmniejszenie zaufania modelu, lub nawet wywoływanie błędów predykcji.

Dla obserwatora z zewnątrz oba te rodzaje zakłóceń mogą zdawać się niezwykle podobne. Jednakże nawet drobne fluktuacje mogą znacząco wpływać na proces decyzyjny [5, 6]. Rozpoznanie, czy dane zostały zakłócone w sposób losowy, czy też kontradydaktryjny, wymaga dość głębokiego zrozumienia ich wpływu na modele neuronowe i rozróżnania różnic pomiędzy nimi. Taka ekspertyza może pomóc w opracowaniu skutecznych technik detekcji oraz strategii odpornościowych.

## **1.1. Cel pracy**

Celem pracy jest empiryczna analiza wpływu zakłóceń, w szczególności szumów losowych oraz celowych zaburzeń kontradyktoryjnych, na modele neuronowe w zadaniach klasyfikacji obrazów. Przeprowadzone zostaną badania, ukierunkowane na zrozumienie, w jaki sposób różne formy zakłóceń oddziałują na zachowanie, efektywność obliczeniową oraz skuteczność modeli neuronowych. Na podstawie tychże badań podjęta zostanie próba do wykorzystania zakłóceń o naturze stochastycznej zarówno do uskuteczniania ataków kontradyktoryjnych, jak i opracowania strategii ich przeciwdziałania.

## **1.2. Zakres pracy**

W ramach pracy planuje się realizację szeregu etapów w celu osiągnięcia wyznaczonych celów. Pierwszym etapem będzie przegląd literatury z zakresu modelowania neuronowego. Następnie przewiduje się przeprowadzenie dyskusji dotyczącej podstawowych teoretycznych aspektów zakłóceń kontradyktoryjnych. Kolejnym krokiem będzie implementacja wybranych modeli neuronowych, po czym przeprowadzone zostaną badania symulacyjne przy różnych typach zakłóceń. Następnie przeprowadzona zostanie analiza uzyskanych rezultatów, a praca zakończy się podsumowaniem wykonanych działań.

## Rozdział 2

# Problematyka zakłóceń w modelowaniu neuronowym

W niniejszym rozdziale zostanie przedstawiony oraz omówiony aktualny stan wiedzy dotyczący podatności modeli neuronowych na zakłócenia w danych wejściowych, opierając się na wybranych publikacjach naukowych. Przeprowadzona zostanie dyskusja nad problematyką zakłóceń w kontekście modelowania neuronowego. Następnie, na podstawie analizy istniejących badań, zarysowany zostanie aktualny stan wiedzy w tej dziedzinie. Stanowić to będzie fundament teoretyczny oraz wstęp do dalszych rozważań.

### 2.1. Źródła zakłóceń

Świat, w którym przyszło nam egzystować, pełen jest chaosu i niedoskonałości. W świecie, gdzie zmienność i niepewność są nieodłącznymi elementami, nawet najbardziej zaawansowane systemy sztucznej inteligencji muszą zmierzyć się z wyzwaniami, jakie stawiają przed nimi nieprzewidywalne w swej naturze **zakłócenia stochastyczne**. W kontekście modelowania neuronowego są to wszelkie perturbacje w danych wejściowych modelu, które pojawiają się za sprawą czynników losowych. Przyczyny tych zakłóceń stochastycznych mogą być różnorodne i często występują łącznie. Oto kilka typowych przykładów:

1. **Czynniki środowiskowe** – środowisko, w którym działa model, może ewoluować w czasie. Przykładowo, zmiany warunków otoczenia mogą obejmować fluktuacje poziomu hałasu tła, zmiany w warunkach oświetleniowych, interferencji elektrycznych oraz wahania temperatury [7]. Wprowadzi to potencjalne niespójności w danych, prowadząc do ich niejednorodności. Przykładowo, w przypadku zbiorów danych pozyskiwanych w skrajnie zróżnicowanych warunkach środowiskowych, zaobserwować można brak spójności, co w konsekwencji prowadzi do problemów z klasyfikacją.

2. **Niedokładności pomiarowe** – dane często są gromadzone przy użyciu różnorodnych narzędzi pomiarowych, takich jak czujniki, kamery oraz mikrofony. Błędy pomiarowe wynikają z niedoskonałości tychże urządzeń, ich ewentualnej niestabilności, stopnia zużycia lub błędów popełnianych przez ludzi w trakcie obsługi [8]. Na przykład, w przypadku kamery cyfrowej, niedoskonałości mogą wynikać z niedokładnej kalibracji, występowania zakłóceń elektrycznych lub mechanicznych, a także błędów ludzkich.
3. **Przetwarzanie i transmisja danych** – w trakcie procesu przetwarzania danych, obejmującego m.in. etapy filtracji, kompresji, konwersji czy kwantyzacji [9], istnieje ryzyko wystąpienia artefaktów lub nawet utraty fragmentów danych. W wypadku danych sygnałowych, takich jak sygnały dźwiękowe, obrazowe czy też biomedyczne, zakłócenia sygnału mogą wynikać z przesłuchów, zakłóceń elektromagnetycznych czy też mechanicznych.

Te pozorne drobne zakłócenia mogą mieć druzgocący wpływ na wydajność modeli neuronowych. Mogą prowadzić do błędnych klasyfikacji, nieprawidłowych przewidywań i niestabilnych wyników. W niektórych przypadkach mogą nawet całkowicie uniemożliwić działanie modelu.

Istnieje jeszcze inna klasa zakłóceń, która zyskuje coraz większe znaczenie w kontekście modelowania neuronowego — są to tzw. **zakłócenia kontradyktoryjne** (ang. *adversarial attacks*). Są to celowo wprowadzone modyfikacje danych wejściowych, które mają na celu zmylenie modelu w taki sposób, aby ten dokonał błędnej interpretacji lub klasyfikacji danych. Zakłócenia te są szczególnie niebezpieczne z uwagi na to, że często są one trudne do wykrycia przez standardowe metody oceny jakości modelu. Typowe metody testowania modeli skupią się głównie na ocenie ogólnej wydajności w zadaniach klasyfikacyjnych, nie biorąc pod uwagę możliwości istnienia specyficznych słabych punktów, które mogą być eksplotowane przez adwersarza. Dodatkowo charakterystyczne dla ataków z wykorzystaniem zakłóceń kontradyktoryjnych jest to, że mogą być one przeprowadzane nawet przy niewielkich zmianach w danych wejściowych, co dodatkowo utrudnia ich wykrycie i obronę przed nimi.

Przykładowo, w zadaniach klasyfikacji obrazów, zakłócenia kontradyktoryjne mogą przybrać formę subtelnych, lecz celowo dobranych zmian w pikselach, mających na celu wprowadzenie modelu w błąd i skłoniienie go do dokonania błędnej klasyfikacji. Perturbacje te mogą być pozornie podobne do tych wywoływanych przez zakłócenia o naturze stochastycznej, co czyni je trudnymi do wykrycia zarówno przez ludzkiego obserwatora, jak i przez standardowe metody przetwarzania obrazu. Podobnie, w kontekście analizy sygnałów dźwiękowych, zakłócenia kontradyktoryjne mogą zostać wprowadzone poprzez manipulację parametrami dźwięku w taki sposób, który przekłamuje jego rzeczywistą treść lub interpretację.

Podsumowując, identyfikacja i efektywne zarządzanie zarówno zakłóceniami stochastycznymi, jak i kontradyktoryjnymi są istniejącymi i wartymi uwagą zagadnieniami w obszarze modelowania neuronowego. Wpływ tych czynników na wydajność modeli jest znaczący, a wykrycie oraz ochrona przed nimi stanowią istotne wyzwania. W obliczu szybkiego postępu w dziedzinie sztucznej inteligencji świadomość tych zagrożeń staje się niezbędna dla zapewnienia bezpieczeństwa i skuteczności systemów opartych na uczeniu maszynowym.

## 2.2. Zakłócenia stochastyczne

Wierne odwzorowanie zjawisk fizycznych, prowadzących do powstawania zakłóceń jest zadaniem niezwykle trudnym ze względu na złożoność natury tych zjawisk oraz obecność różnorodnych czynników wpływających na ich przebieg. Dlatego wykorzystanie szumów generowanych za pomocą określonych rozkładów zmiennej losowej staje się niezwykle przydatnym narzędziem badawczym. Umożliwia to stworzenie kontrolowanych warunków eksperymentalnych, co z kolei pozwala na reprodukcję i analizę różnorodnych zakłóceń w sposób precyzyjny i powtarzalny. Szumy generowane za pomocą rozkładów losowych otwierają możliwość symulacji różnych scenariuszy oraz manipulacji parametrami zakłóceń. Przykładowe rodzaje szumów, które mogą być wykorzystane do modelowania rzeczywistych zakłóceń w badaniach to:

1. **Szum biały o rozkładzie jednostajnym** [9] – wartości są równomiernie rozłożone na określonym przedziale, tj. pochodzą z rozkładu jednostajnego, a kolejne próbki są od siebie niezależne.
2. **Szum biały o rozkładzie normalnym** [10] – przyjmuje wartości o rozkładzie normalnym, co oznacza, że ich prawdopodobieństwo jest największe wokół średniej, a maleje wraz z oddalaniem się od niej, zgodnie z klasycznym kształtem krzywej Gaussa.
3. **Szum Cauchy'ego** [11] – charakteryzuje się brakiem wartości oczekiwanej, co oznacza m.in.: że jego wariancja jest niezdefiniowana. Jest to rodzaj szumu szczególnie podatny na sporadyczne skrajne odstępstwa pojedynczych perturbacji od średniej arytmetycznej szumu, co sprawia, że jest użyteczny w analizie danych odstających.

### 2.2.1. Wpływ szumu na klasyfikację obrazów

Istnieje szereg prac poświęconych tematyce badania wpływu wszelkiego rodzaju niezłożliwości zaburzeń na obrazach, na proces ich klasyfikacji. W literaturze można znaleźć liczne badania analizujące, jak szumy, artefakty kompresji lub inne zniekształcenie mogą wpływać na skuteczność systemów klasyfikacyjnych. Badania w tym zakresie często wykorzystują różnorodne zestawy danych oraz symulacje, aby ocenić i porównać ich oddziaływanie w zależności od parametrów modelu, rozmiaru zdjęć, stopnia zniekształcenia danych etc.

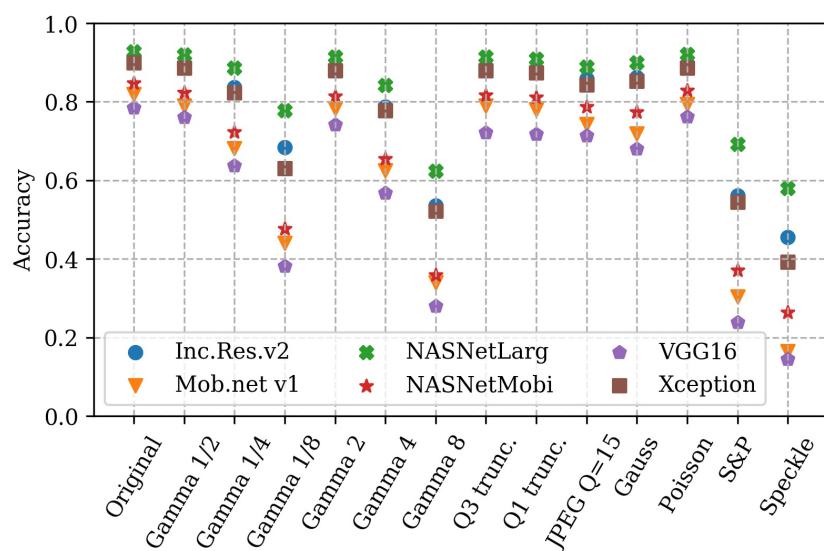
W pracy zatytułowanej *Can Exposure, Noise and Compression affect Image Recognition?* [12] autorzy podjęli próbę szczegółowego zrozumienia wpływu trzech głównych typów degradacji jakości obrazu – ekspozycji, kompresji oraz szumu – na wydajność systemów rozpoznawania obrazów. Autorzy zauważają, że w realnych zastosowaniach, obrazy wykorzystywane przez systemy komputerowe często podlegają różnym formom zniekształceń. Motywacją badaczy była chęć zidentyfikowania, które z tych zniekształceń mają największy wpływ na dokładność modeli rozpoznawania obrazów, co pozwoli na opracowanie bardziej odpornych systemów.

Badania zostały przeprowadzone przy użyciu kilku standardowych zestawów danych, m.in: **Open Images**, **CIFAR** oraz popularnych modeli głębokich sieci neuronowych, m.in: **MobileNetV1**, **ResNet** czy **VGG**. W celu analizy wpływu degradacji, autorzy wprowadzali zmiany w ekspozycji, poziomie zaszumieniem szumami o rozkładzie **Gauss'a**, **Poisson'a**, **szumem ziarnistym** (ang. *salt-and-pepper noise*) oraz **szumem plamkowym** (ang. *speckle noise*). Badaniom poddano także różne poziomy kompresji formatu **JPEG**. Modele zostały następnie ocenione pod kątem ich wydajności na tych zdegradowanych obrazach, a wyniki zostały porównane z wynikami uzyskanymi na obrazach niezmodyfikowanych.



Rys. 2.1: Przykładowy podgląd badanych zniekształceń. Od lewej: kompresja JPEG, szum ziarnisty, szum o rozkładzie Gauss'a, szum o rozkładzie Poisson'a, szum plamkowy [12]

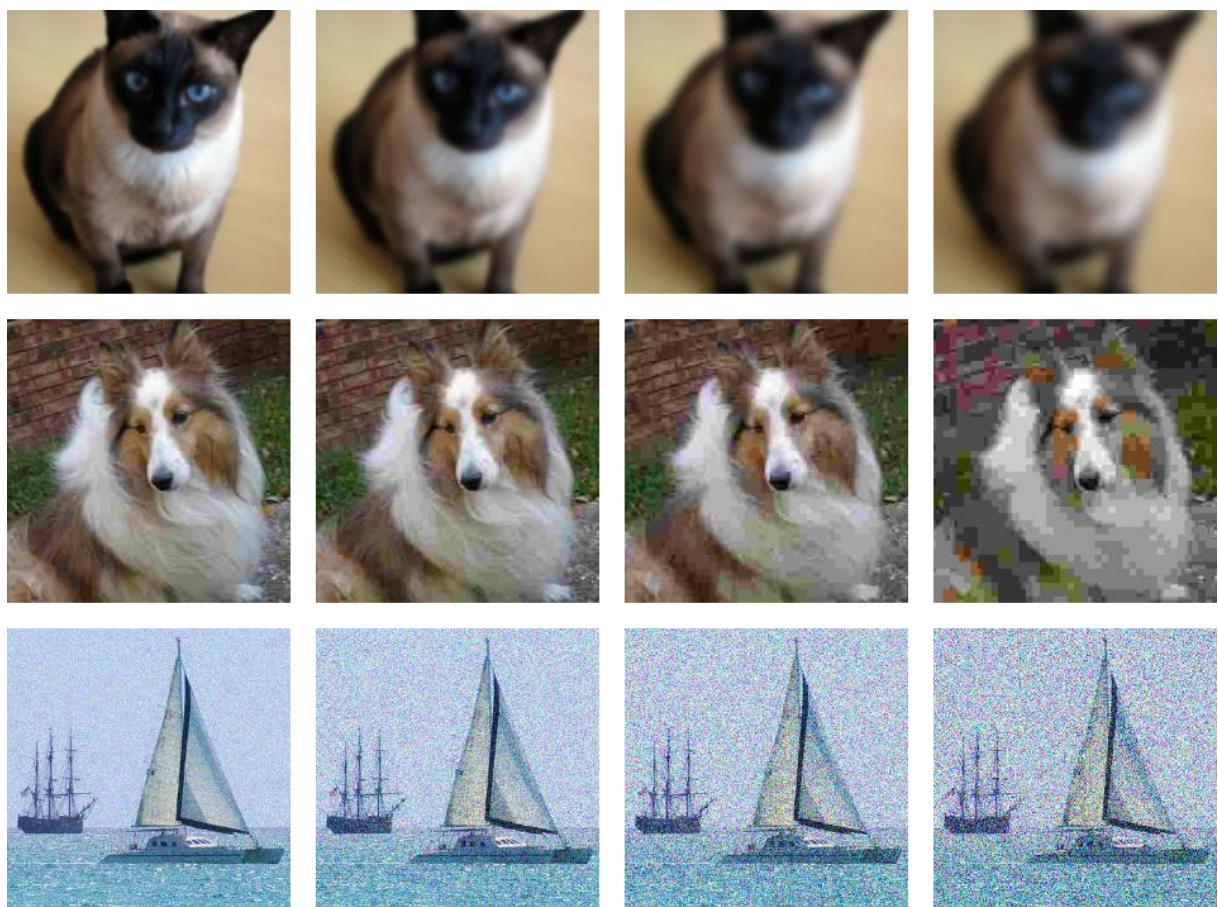
Wyniki badań wykazały, że wszystkie testowane modele wykazywały znaczną wrażliwość na zmiany jakości obrazu. Największy wpływ na dokładność rozpoznawania miały szum i rozmycie, które powodowały istotne spadki wydajności. Oceniono, iż szum o rozkładzie Gauss'a oraz o rozkładzie Poisson'a wykazują minimalny wpływ na dokładność większości ocenianych modeli. Zauważono jednak większą wrażliwość na zniekształcenia, które nie są skorelowane z oryginalnym obrazem, zwłaszcza na szum ziarnisty oraz plamkowy. Ekspozycja i kompresja miały również wpływ na wyniki, choć w mniejszym stopniu. Autorzy wskazują, że istnieje pilna potrzeba rozwijania bardziej odpornych modeli, które będą w stanie efektywnie radzić sobie z obrazami o obniżonej jakości, co jest newralgiczne dla ich zastosowań w rzeczywistych scenariuszach.



Rys. 2.2: Zbiorcze podsumowanie wyników badań [12]

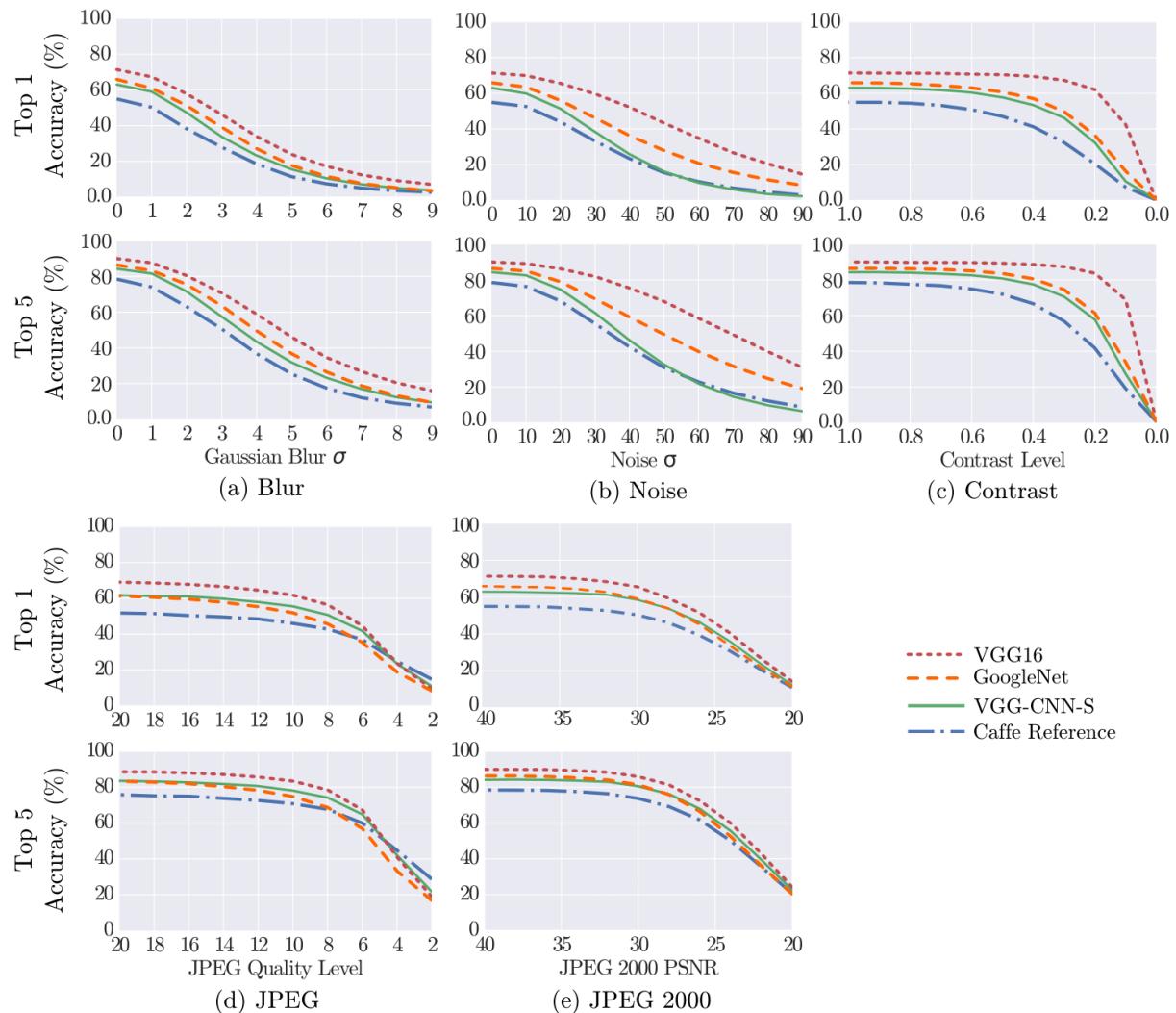
W pracy zatytułowanej *Understanding How Image Quality Affects Deep Neural Networks* [13] jest kompleksowym studium wpływu jakości obrazu na funkcjonowanie głębszych sieci neuronowych. Autorzy skoncentrowali się na analizie wpływu zakłóceń, takich jak kompresja, rozmycie, oraz szumy, na dokładność klasyfikacji przez modele neuronowe. Motywacją autorów wynikała z potrzeby głębszego zrozumienia, jak modele neuronowe radzą sobie z obrazami o obniżonej jakości. W rzeczywistych aplikacjach, takich jak monitorowanie wizyjne, analiza obrazów medycznych czy systemy autonomiczne, obrazy często nie są idealne. Autorzy zauważali, iż wiele dotychczasowych badań koncentrowało się na trenowaniu i testowaniu modeli na wysokiej jakości obrazach, co nie zawsze odzwierciedla rzeczywiste warunki pracy tych systemów. Celem autorów było zbadanie, jak różne poziomy jakości obrazu wpływają na wydajność sieci neuronowych, co mogłoby przyczynić się do opracowania bardziej odpornych modeli.

Do przeprowadzenia badań autorzy wybrali cztery reprezentatywne modele sieci neuronowych: **Caffe Reference**, **VGG-CNN-S**, **VGG-16** oraz **GoogleNet**. Dane użyte do eksperymentów pochodziły z zestawu **ILSVRC 2012** (ImageNet). Zestaw obrazów został zniekształcony przy użyciu czterech metod: kompresji, rozmycia, zaszumienia oraz zmiany kontrastu. Każda z tych metod została zastosowana w różnych intensywnościach, aby dokładnie zbadać, jak różne poziomy zakłóceń wpływają na wydajność sieci neuronowych. Rozmycie zostało osiągnięte za pomocą **filtrów Gauss'a**, szumy zostały dodane na podstawie **rozkładu Gauss'a**. Jako algorytm kompresji badano format **JPEG** oraz jego unowocześnioną wersję **JPEG 2000**.



Rys. 2.3: Przykładowy podgląd badanych zniekształceń. Od góry: rozmycie, kompresja, szum [13]

Wyniki przeprowadzonych eksperymentów jednoznacznie wskazują, że jakość obrazu ma istotny wpływ na dokładność klasyfikacji przez sieci neuronowe. Wszystkie analizowane modele wykazały spadek wydajności w odpowiedzi na wzrastający poziom zakłóceń. Zastosowanie kompresji obrazu prowadziło do znacznej utraty istotnych szczegółów, co w efekcie powodowało znaczący spadek dokładności klasyfikacji, lecz dopiero dla względnie dużych poziomów kompresji. Modele neuronowe miały szczególne trudności z rozpoznawaniem cech w obrazach skompresowanych do formatu JPEG. Wykazano, że rozmycie wpływało głównie na późniejsze warstwy konwolucyjne, gdzie ekstrakcja złożonych cech była utrudniona. Mimo że wczesne warstwy sieci nie były tak silnie dotknięte, ogólna wydajność sieci znacząco spadała. Szum natomiast miał wysoce destruktywny wpływ na wczesne warstwy konwolucyjne, prowadząc do wielu nieprawidłowych aktywacji. Propagacja tych nieprawidłowości przez kolejne warstwy sieci znacząco obniżała dokładność klasyfikacji. Szum okazał się jednym z najbardziej destrukcyjnych zakłóceń.



Rys. 2.4: Zbiorcze podsumowanie wyników badań [13]

## 2.2.2. Obszary wymagające dalszych badań

Po przeprowadzeniu starannej analizy literatury naukowej dotyczącej wpływu zakłóceń na proces klasyfikacji obrazów w modelowaniu neuronowym nie odnaleziono żadnej pracy, która skupiałaby się specyficznie na badaniu wpływu szumu białego o rozkładzie jednostajnym oraz szumu Cauchy'ego na ten proces. Choć istnieją liczne publikacje poświęcone problemowi zakłóceń w danych wejściowych modeli klasyfikacji obrazów, analiza tych materiałów nie ujawniła żadnej pracy, która bezpośrednio poruszałaby tę kwestię. Jest to istotna luka w obecnej literaturze, która wskazuje na potrzebę dalszych badań nad tym zagadnieniem, mających na celu lepsze zrozumienie wpływu tych szumów na skuteczność procesu klasyfikacji obrazów, oraz potencjalne metody ich zwalczania.

## 2.3. Zakłócenia kontradyktoryjne

W kontekście modelowania neuronowego identyfikuje się trzy główne typy ataków kontradyktoryjnych: **atak zatruwający** (ang. *poisoning attack*), **atak unikający** (ang. *evasion attack*), oraz **atak eksploracyjny** [14]. (ang. *exploratory attack*). Pierwsze dwie metody skupią się na wprowadzeniu zakłóceń kontradyktoryjnych do danych zbioru treningowego (atak zatruwający) lub testowego (atak unikający) w taki sposób, aby zagrozić całemu procesowi uczenia. Natomiast techniki ataków eksploracyjnych w ogóle nie wpływają na zbiory danych szkoleniowych, lecz operują na już przetrenowanym modelu.

Uzyskanie dostępu do modelu, a następnie obserwacja jego reakcji na różnorodne dane wejściowe pozwala adwersarzowi na zrozumienie zasad działania algorytmu i odkrycie ewentualnych wzorców błędów czy słabości modelu. Stopień zaawansowania ataku zależy od poziomu informacji, którymi atakujący dysponuje na temat struktury modelu, parametrów, czy nawet samego zestawu danych treningowych. Następnie, na bazie zgromadzonych danych, adwersarz generuje perturbacje na nowych danych wejściowych modelu, których celem jest zmniejszenie zaufania modelu lub wywołanie błędnych wyników klasyfikacji. W niniejszej pracy skupiono się właśnie na tym rodzaju zakłóceń kontradyktoryjnych.

### 2.3.1. Atak *FGSM*

W pracy zatytułowanej *Explaining and harnessing adversarial examples* [1] autorzy skupili się na problematyce zakłóceń kontradyktoryjnych, w szczególności na zrozumieniu natury przykładów kontradyktoryjnych. Autorzy pracy postawili hipotezę, że jedną z głównych przyczyn podatności modeli neuronowych na zakłócenia kontradyktoryjne jest ich liniowość. Nawet w złożonych nieliniowych architekturach, zachowanie modeli w wysokowymiarowej przestrzeni cech jest zbliżone do liniowego. W pracy wykazano, że problem ten jest szczególnie wyraźny w modelach głębokiego uczenia, gdzie liczba wymiarów jest bardzo duża, co w praktyce umożliwia efektywne generowanie niewielkich, acz wysoce złośliwych zakłóceń.

Zasadniczym elementem pracy jest wprowadzenie algorytmu **FGSM** (ang. *Fast Gradient Sign Method*), który służy do szybkiego generowania zakłóceń kontradyktoryjnych. Algorytm ten wykorzystuje gradient funkcji koszta  $\omega$  w stosunku do obrazu wejściowego  $x$ , aby stworzyć niewielkie perturbacje, które docelowo prowadzić mają do błędnej klasyfikacji, lub do znacznego obniżenia zaufania modelu. Jak pokazano w pracy *Black-box Adversarial Sample Generation Based on Differential Evolution* [15], wariacja algorytmu o nazwie BMI-FGSM może być zastosowana w scenariuszach *black-box attack* [16], gdzie atakujący nie ma dostępu do architektury ani pełnej informacji o modelu.

Podstawowa wersja algorytmu, a więc dla scenariusza *white-box attack* [16], dla danego modelu z parametrami  $\theta$  wymaga obliczenia gradientu funkcji straty  $\nabla J(\theta, x, y)$ , gdzie  $y$  to faktyczna etykieta przypisana do obrazu wejściowego  $x$ . Poszczególne wartości gradientu wskazują na kierunek najszybszego wzrostu wartości funkcji straty w przestrzeni cech. Następnie wyznacza się perturbację  $\delta$ , która zostanie dodana do oryginalnego przykładu  $x$ , tworząc tym samym przykład kontradyktoryjny  $x'$ . Perturbacja ta jest określona przez znak gradientu funkcji straty, a jej wielkość skalowana jest przez mały współczynnik  $\epsilon$ . Ogólny wzór na wynikowe zaburzenie  $\delta$  przedstawia się następująco:

$$\delta = \epsilon \cdot |\nabla_x J(\theta, x, y)|$$

gdzie:

$\nabla_x J$  – gradient funkcji straty

$\theta$  – parametry modelu

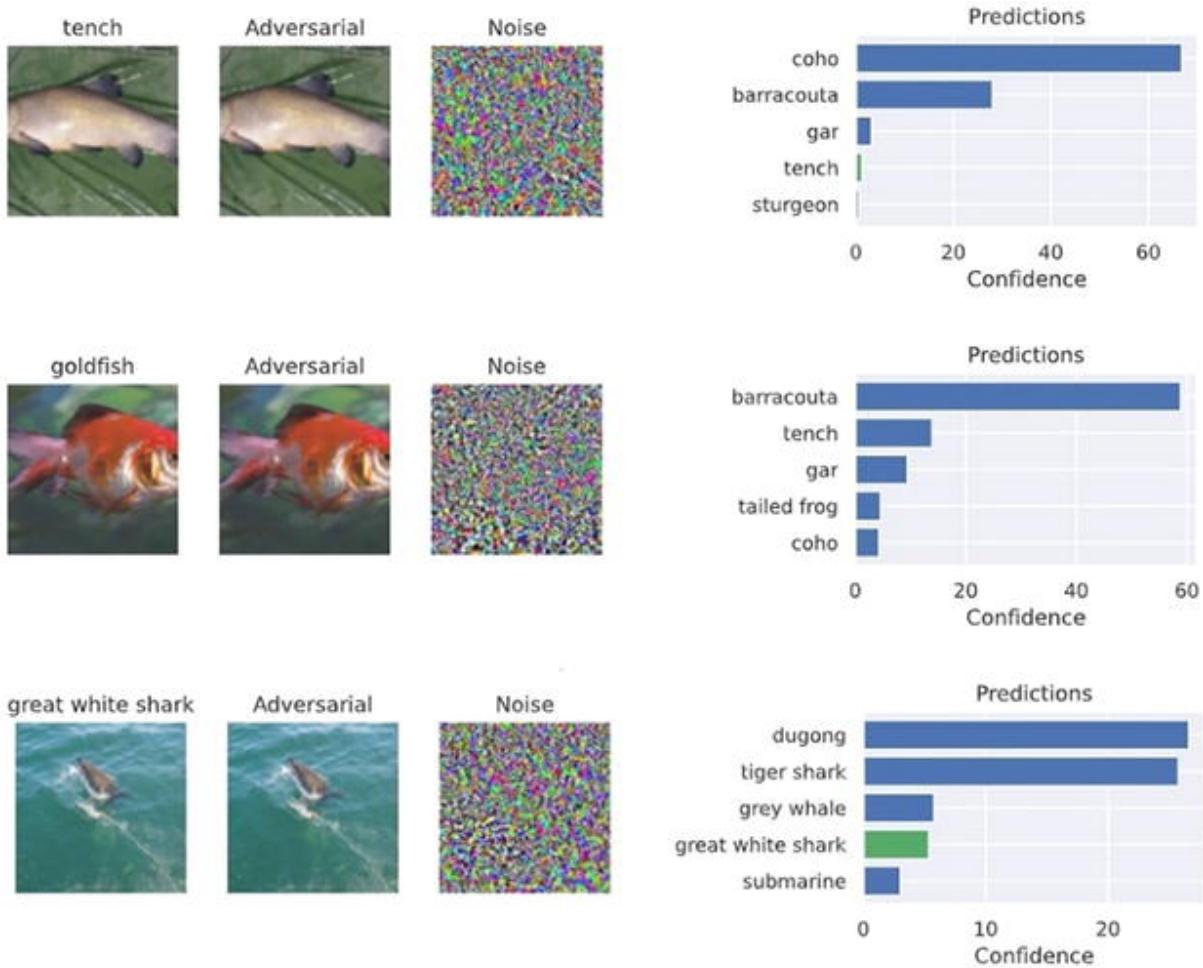
$x$  – oryginalny obraz wejściowy

$y$  – etykieta obrazu wejściowego

$\epsilon$  – współczynnik wzmocnienia

$\delta$  – wynikowe zakłócenie

Skuteczność algorytmu zbadali autorzy pracy *Adversarial attacks on Image classification models: FGSM and patch attacks and their impact* [17]. W badaniu wykorzystano trzy pretrained modele konwolucyjne: **ResNet-34**, **GoogleNet** oraz **DenseNet-161**. Każdy z tych modeli został wybrany ze względu na swoją popularność i sprawdzoną skuteczność w zadaniach klasyfikacji obrazów. Do oceny wpływu zakłóceń kontradyktoryjnych użyto obrazów z publicznie dostępnego zbioru danych **ImageNet**. Główną motywacją autorów było zbadanie, jak zakłócenia kontradyktoryjne wpływają na dokładność klasyfikacji obrazów przez zaawansowane modele konwolucyjne oraz chęć zidentyfikowania słabych punktów w popularnych modelach CNN i opracowanie strategii obronnych, które mogłyby poprawić ich odporność na takie ataki. Autorzy przeprowadzili szczegółową analizę porównawczą wyników klasyfikacji przed i po zastosowaniu zakłóceń przy zastosowaniu metryki **top 5 Accuracy**.



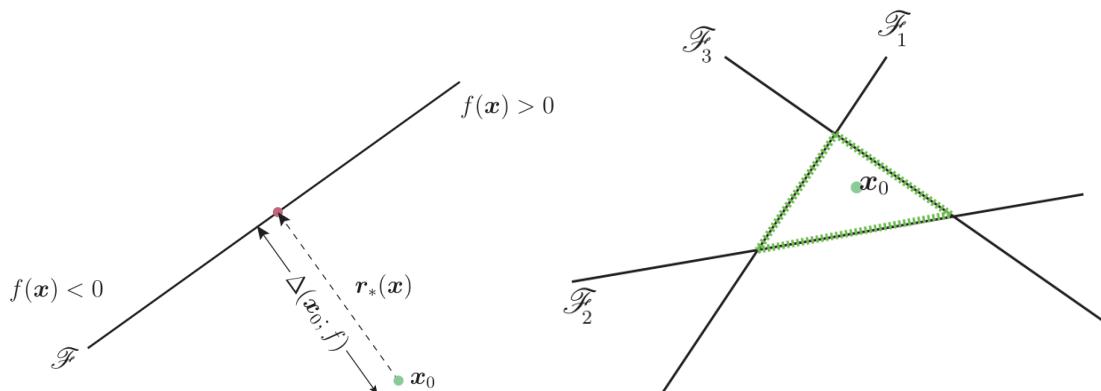
Rys. 2.5: Przykładowy podgląd zakłóceń oraz wyników klasyfikacji [17]

Wyniki jednoznacznie wskazują, że algorytm FGSM znaczco obniża dokładność klasyfikacji obrazów przez badane modele CNN. Wzrost wartości  $\epsilon$  powodował systematyczne zwiększenie błędów klasyfikacji, co potwierdza wysoką wrażliwość modeli na zakłócenia kontradydaktryjne. Dla parametru  $\epsilon = 0.02$ , co oznaczał zmianę wartości pikseli o około 1 w skali 0 – 255, żaden z modeli nie był w stanie poprawnie sklasyfikować obrazów zakłóconych. Wartość ta jest tak mała, że zmodyfikowany obraz jest wizualnie nieodróżnialny od oryginalnego. Dodatkowo autorzy zauważyli, że podatność modeli może być związana z architekturą modelu oraz sposobem, w jaki modele te uczą się reprezentacji danych.

### 2.3.2. Atak *DeepFool*

W pracy zatytułowanej *DeepFool: a simple and accurate method to fool deep neural networks* [2] autorzy skupili się na problematyce stabilności modeli neuronowych wobec niewielkich perturbacji w obrazach, które mogą prowadzić do błędnych klasyfikacji. W pracy tej zaproponowano i szczegółowo omówiono algorytm **DeepFool**, który jest zaprojektowany do generowania minimalnych perturbacji, które jednocześnie powinny być wystarczające, aby doprowadzić do błędnej klasyfikacji, lub do znacznego obniżenia zaufania modelu.

Ogólnym celem algorytmu jest znalezienie minimalnego zaburzenia  $r^*(x)$ , które sprawi, że klasyfikator  $f$  błędnie sklasyfikuje zaburzony obraz wejściowy  $x'$ . Aby osiągnąć ten cel, zaburzenie  $r^*(x)$  musi być wystarczające, aby przesunąć przewidywaną etykietę  $f(x)$  obrazu  $x'$  ortogonalnie do hiperpłaszczyzny (granicy decyzyjnej)  $\mathcal{F}$  klasyfikatora binarnego, lub względem najbliższej z płaszczyzn  $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \dots, \mathcal{F}_n$  dla klasyfikatora  $n$  klasowego. W praktyce algorytm DeepFool dla klasyfikatorów wieloklasowych rozszerza podejście klasyfikatora binarnego, traktując klasyfikator wieloklasowy jako zbiór wielu klasyfikatorów binarnych. Granicę decyzyjną klasyfikatora wieloklasowego można traktować jako wielościan utworzony przez przecięcie wielu hiperpłaszczyzn.



Rys. 2.6: Schemat układu hiperpłaszczyzn (granic decyzyjnych). Od lewej: dla klasyfikatora binarnego, dla klasyfikatora wieloklasowego [2]

Algorytm oblicza aktualne minimalne zaburzenie  $r^*(x_i)$  w iteracji  $i$ , dzieląc wynik predykcji  $f(x_{i-1})$  przez normę  $L2$  obliczonego gradientu  $\omega$  funkcji straty, co daje skalarną wartość zaburzenia. Następnie ta wartość jest mnożona przez wektor jednostkowy  $\omega$  z wykorzystaniem normy  $L2$ . Całość mnożona jest przez liczbę  $-1$ , aby docelowo zwiększyć stratę klasyfikatora  $f$ . W praktyce, ze względu na nieliniową naturę rzeczywistych modeli neuronowych, jednorazowa procedura może być dalece niewystarczająca, aby doprowadzić do błędu klasyfikacji lub pożądanego obniżenia zaufania modelu. Aby temu zaradzić, algorytm działa iteracyjnie, dodając aktualne zaburzenie  $r^*(x_i)$  do uprzedniego obrazu  $x_{i-1}$ , aż do zmiany etykiety lub osiągnięcia maksymalnej liczby iteracji. Dodatkowo, aby zapobiec zakończeniu procesu dokładnie na granicy decyzyjnej, wprowadzono specjalny parametr *overshoot*  $\eta$ , który skaluje zaburzenie  $r^*(x)$  przez wartość  $(1 + \eta)$ , docelowo doprowadzając do przekroczenia granicy decyzyjnej. Ogólny wzór na perturbację  $r^*(x_i)$  w iteracji  $i$  dla klasyfikatora binarnego przedstawia się następująco:

$$r^*(x_i) = -1 \cdot \frac{f(x_{i-1})}{\|\omega\|_2^2} \cdot \omega$$

gdzie:

$r^*(x_i)$  – wynikowa perturbacja w iteracji  $i$

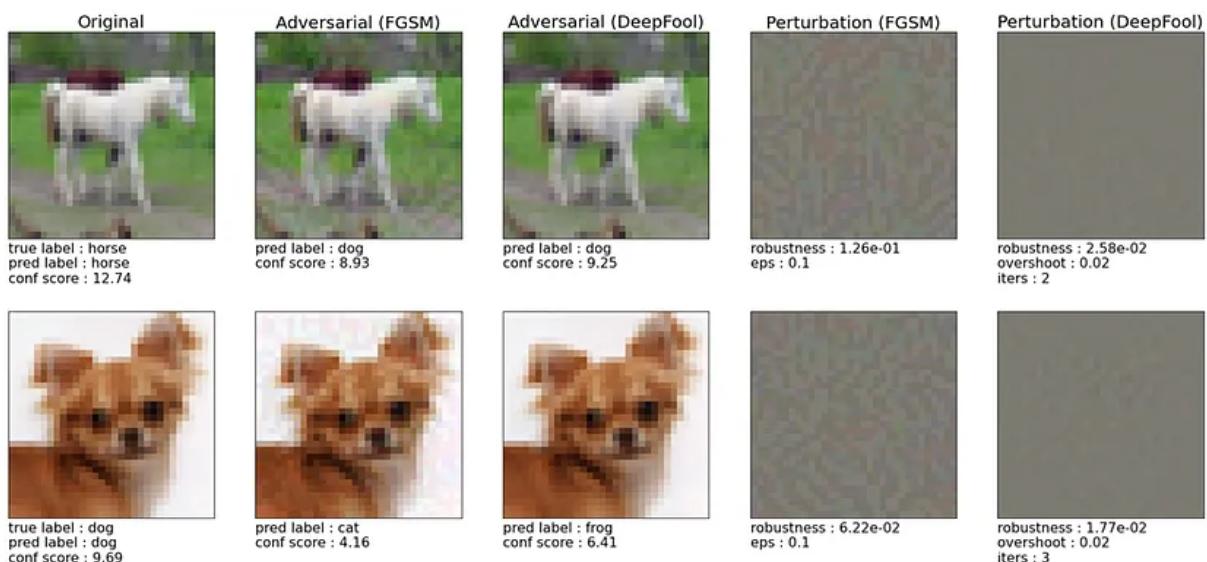
$f(x_{i-1})$  – wynikowa predykcja w iteracji  $i - 1$

$\omega$  – aktualny gradient funkcji straty

W przypadku problemów klasyfikacji wieloklasowej gradient funkcji straty oraz propagacja wsteczna muszą być obliczone dla każdej klasy z osobna. Do znalezienia minimalnego zaburzenia  $r^*(x_i)$  w danej iteracji  $i$  należy wyznaczyć różnicę między wartościami dla każdej etykiety obrazu z perturbacjami  $x_{i-1}$  a etykietami z oryginalnego obrazu  $x$ . Po określeniu klasy  $l(x_i)$ , która jest najbliższą jednej z hiperpłaszczyzn  $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \dots, \mathcal{F}_n$  względem aktualnego punktu predykcji  $f(x_i)$ , oblicza się aktualne zaburzenie  $r^*(x_i)$  w bieżącej iteracji. Podobnie jak w przypadku klasyfikatora binarnego, po obliczeniu zakłócenia  $r^*(x_i)$  zostanie ono dodane do uprzedniego obrazu  $x_{i-1}$ . W przypadku klasyfikatora wieloklasowego również stosuje się skalar przekroczenia  $(1 + \eta)$ , a cała procedura powtarzana jest aż do zmiany etykiety lub osiągnięcia maksymalnej dozwolonej liczby iteracji.

W artykule *A Review of DeepFool: a simple and accurate method to fool deep neural networks* [18] dokonano szczegółowej analizy oraz porównania wpływu algorytmów DeepFool i FGSM na efektywność modeli neuronowych w zadaniach klasyfikacji obrazów. Eksperymenty przeprowadzono, wykorzystując trzy główne zbiory danych: **MNIST**, **CIFAR-10** oraz **ILSVRC2012**. W przypadku zbioru danych ILSVRC2012 zastosowano model **GoogLeNet**, zaawansowany model głębokiej sieci konwolucyjnej, który osiągnął wysokie wyniki w konkursie ILSVRC2014. Do analiz zbioru danych MNIST wybrano architektury **LeNet** oraz **FC-150-50**, natomiast dla zbioru CIFAR-10 wykorzystano modele **LeNet** oraz **NIN**.

Wyniki analizy pokazują, że DeepFool generuje bardziej subtelne, acz precyzyjne perturbacje, co skutkuje wyższą precyją w generowaniu ataków. Dzięki temu mają one większy potencjał do skutecznego zmieniania etykiety klasyfikacji, lecz okupione to jest kosztem większej złożoności obliczeniowej iteracyjnego charakteru całej procedury. FGSM, z kolei, działa szybciej, generując perturbacje w jednej iteracji, co jest korzystne w kontekście efektywności czasowej. Okupione to jest jednak większymi, bardziej widocznymi zakłóceniami, które mogą być łatwiej wykrywane przez obserwatora ludzkiego.



Rys. 2.7: Przykładowy podgląd porównawczy zakłóceń [18]

## 2.4. Propozycja zakłócenia kontradyktoryjnego indukowanego stochastycznie

Algorytm *Low Energy Stochastic Gradient Sign Method* (LESGSM) jest autorską wariacją algorytmu FGSM, zaprojektowaną do generowania zakłóceń kontradyktoryjnych. Atak LESGSM rozszerza FGSM poprzez wprowadzenie mechanizmów stochastycznych oraz iteracyjnej optymalizacji, mających na celu minimalizację energii perturbacji na przykładzie kontradyktoryjnym. Fundamentalnym celem LESGSM jest znalezienie bardziej subtelnych i trudniejszych do wykrycia perturbacji, które nadal skutecznie wprowadzają model w błąd.

LESGSM wykorzystuje funkcję straty *Sparse Categorical Crossentropy*, która jest standar-dową miarą odchylenia predykcji modelu od rzeczywistych etykiet w zadaniach klasyfikacji wieloklasowej. Funkcja ta oblicza entropię krzyżową między rzeczywistymi etykietami a przewidywianymi prawdopodobieństwami klas, co pozwala na ocenę, jak dobrze model przewiduje prawidłowe klasy. Algorytm w pierwszej swej fazie oblicza gradient funkcji straty względem wejściowych obrazów. Gradient ten wskazuje kierunek największego wzrostu straty, a jego znak jest używany do stworzenia początkowych perturbacji, które są skalowane przez parametr  $\epsilon$ . Skalowanie to kontroluje wielkość perturbacji, umożliwiając dostosowanie ich intensywności.

Algorytm przetwarza obrazy w partiach (ang. *batches*), co jest typową praktyką w uczeniu maszynowym ze względu na ograniczenia pamięciowe i efektywność obliczeniową. Dla każdej partii przeprowadzany jest iteracyjny proces optymalizacji perturbacji:

- Obliczenia aktualnych wskaźników** – Na początku każdej partii algorytm oblicza predykcje modelu i stratę dla obrazów z początkowymi perturbacjami. Obliczana jest również energia perturbacji, która jest sumą kwadratów wartości perturbacji. Energia ta służy jako miara intensywności perturbacji i będzie optymalizowana w kolejnych iteracjach.
- Optymalizacja stochastyczna** – W każdej iteracji partii, algorytm modyfikuje perturbacje poprzez mnożenie ich przez losowe wartości z zakresu  $\forall \xi \in [0.97, 1.02]$ . Parametry te zostały dobrane w ten sposób, aby wartość oczekiwana była mniejsza niż 1, a więc aby docelowo globalnie zmniejszać energię perturbacji. Dokładną granicę dolną  $\xi_{min} = 0.97$  oraz granicę górną  $\xi_{max} = 1.02$  wyznaczono w sposób eksperymentalny dla danych wejściowych z zakresu  $\forall X \in [0, 1]$ .
- Obliczenia wtórnego wskaźników** – W każdym kroku iteracji algorytm generuje nowe perturbacje przez stochastyczne modyfikowanie aktualnych perturbacji. Obliczane są predykcje modelu i strata dla nowych perturbacji, a także energia tych perturbacji. Dodatkowo algorytm oblicza wartość tolerancji odporności ang. (*robust tolerance*) dla partii, która jest funkcją logarytmiczną straty partii skalowanej przez parametr  $v$ . Jeżeli energia nowych perturbacji jest mniejsza niż energia aktualnych perturbacji, a strata nowych perturbacji jest większa niż wartość tolerancji odporności, aktualne perturbacje są zastępowane nowymi perturbacjami.

4. **Dyskredytacja perturbacji wtórnych o energii dodatniej** – Po zakończeniu iteracji dla partii, zmodyfikowane perturbacje są zapisywane, co umożliwia zachowanie aktualnego stanu perturbacji dla kolejnych partii. Na zakończenie wszystkich iteracji dla wszystkich partii algorytm modyfikuje wynikowe perturbacje, aby wyeliminować komponenty o dodatniej energii.

W celu lepszego zrozumienia zasady działania algorytmu LESGSM poniżej zamieszczona jest pseudokod, który ilustruje istotne etapy jego działania. Pseudokod ten pokazuje, jak algorytm wykorzystuje gradienty w celu generowania zakłóceń początkowych, które to następnie iteracyjnie są próbkowane szumem, aby maksymalnie obniżyć ich energię, przy zachowaniu złośliwych cech wynikowych perturbacji.

---

#### **Algorithm 1** Low Energy Stochastic Gradient Sign Method (LESGSM)

---

```

1: Input: Model model, Images images, Labels labels, Perturbation scale  $\epsilon$ , Robustness parameter  $v$ , Batch size batch_size, Number of iterations batch_iter, Clipping range clip
2: Output: Perturbed images
3:
4:  $pert \leftarrow \vec{0}$ 
5:  $loss \leftarrow calculateLoss(labels, model(images))$ 
6:  $grad \leftarrow calculateGradient(loss, images) \cdot \epsilon$ 
7:
8: for each batch do
9:    $batch\_pert \leftarrow pert[batch]$ 
10:   $batch\_pred \leftarrow model(images[batch])$ 
11:   $batch\_loss \leftarrow calculateLoss(labels[batch], batch\_pred)$ 
12:
13:   $batch\_energy \leftarrow \sum(grad[batch]^2)$ 
14:
15:  for each biter do
16:     $biter\_pert \leftarrow batch\_pert * uniform(0.97, 1.02)$ 
17:     $biter\_pred \leftarrow model(clipToRange(images[batch] + biter\_pert, clip))$ 
18:     $biter\_loss \leftarrow calculateLoss(labels[batch], biter\_pred)$ 
19:
20:     $biter\_energy \leftarrow \sum(biter\_pert^2)$ 
21:     $biter\_robust \leftarrow \frac{1}{v} \cdot log(v * batch\_loss + 1)$ 
22:
23:    if  $batch\_energy > biter\_energy$  and  $biter\_loss > biter\_robust$  then
24:       $batch\_energy \leftarrow biter\_energy$ 
25:       $batch\_pert \leftarrow biter\_pert$ 
26:    end if
27:  end for
28:
29:   $pert[batch : batch + batch\_size] \leftarrow batch\_pert$ 
30: end for
31:
32:  $pert \leftarrow sign - clipToRange(sign - pert, [-\infty, 0])$ 
33: return  $clipToRange(images + pert, clip)$ 

```

---

## Rozdział 3

# Metodologia badań

Z uwagi na przeglądowy aspekt pracy, zdecydowano się na wykorzystanie narzędzi o uniwersalnym zastosowaniu, których obsługa nie wymaga specjalistycznej wiedzy. Decyzja ta została podjęta w celu ułatwienia i usprawnienia procesu badawczego oraz zapewnienia dostępu do narzędzi powszechnie używanych w dziedzinie. Dodatkowo, aby maksymalnie uogólnić wyniki badań, wybrano proste zadania z zakresu klasyfikacji obrazów. To podejście pozwoliło na osiągnięcie większej elastyczności w przygotowywaniu prób badawczych i nanoszeniu poprawek w modelach neuronowych, a także ułatwiło interpretację wyników badań.

### 3.1. Środowisko uruchomieniowe

Badania zostały przeprowadzone na urządzeniu pracującym pod kontrolą systemu operacyjnego **Windows 11 Education** w wersji 23H2 niewyposażonego w dedykowany układ graficzny. Na urządzeniu zainstalowane było środowisko wykonawcze **Python** w wersji 3.12.3, obsługujące wirtualne środowisko obliczeniowe **Jupyter** w wersji 2024.4.0 w formie rozszerzenia do edytora kodu **Visual Studio Code** w wersji 1.89.0. Postęp prac dokumentowano za pomocą systemu kontroli wersji **Git** w wersji 2.37.3.windows.1.

### 3.2. Biblioteki programistyczne

Jako główne narzędzie programistyczne wybrano bibliotekę **TensorFlow** w wersji 2.16.1. Zawiera ona w sobie szereg narzędzi, m.in. interfejs **Keras** czy spreparowane zbiory danych. Biblioteka TensorFlow zapewnia elastyczność w projektowaniu różnorodnych architektur modeli neuronowych (klasyczne, konwolucyjne, rekurencyjne), dzięki czemu możliwe jest eksperymentowanie z różnymi rodzajami modeli w ramach jednej platformy. Dostępna jest obszerna dokumentacja, co ułatwia proces poszukiwania informacji i rozwiązania problemów. Z uwagi na brak dedykowanego procesora graficznego niebywałyム atutem jest również optymalizacja pod współczesne procesory o architekturze **x86-64**.

Ponadto biblioteka TensorFlow jest kompatybilna z wieloma innymi narzędziami i framework'ami powszechnie używanymi w obszarze uczenia maszynowego. Posiada również natywne wsparcie dla różnych formatów danych, w tym dla popularnej biblioteki do obsługi wielowymiarowych macierzy **NumPy**. Dzięki temu możliwe było skorzystanie z kilku innych bibliotek:

1. **Matplotlib** - narzędzie służące do tworzenia i obsługi wykresów. Umożliwia generowanie wielu podstawowych typów wykresów i zażądanie nimi.
2. **pandas** - narzędzie do manipulacji danymi, które dostarcza wygodne i efektywne struktury danych, takie jak *DataFrame*, oraz zestaw funkcji do operacji na danych, takich jak filtrowanie, grupowanie, sortowanie, łączenie itp.
3. **seaborn** - narzędzie wizualizacji danych, zbudowane bazie Matplotlib. Oferuje bardziej zaawansowane i estetyczne wykresy, takie jak histogramy, mapy ciepła (ang. *heat maps*).

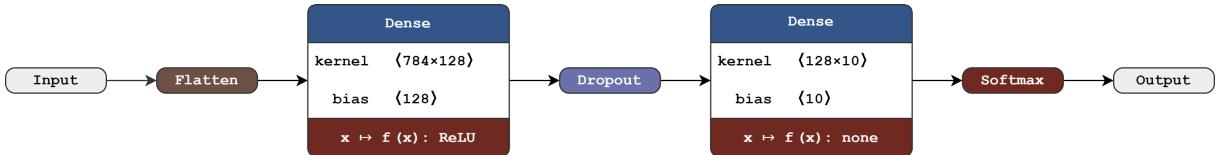
### 3.3. Zbiory danych i modele neuronowe

Do badań wybrano dwa zbiory danych: **MNIST** [19] oraz **Fashion MNIST** [20]. Zbiór danych MNIST jest jednym z najbardziej znanych zbiorów w dziedzinie klasyfikacji obrazów. Zawiera on ręcznie pisane cyfry od 0 do 9, przeskalowane do rozmiaru  $28 \times 28$  pikseli w skali szarości. Z kolei Fashion MNIST to stosunkowo nowy zbiór danych, który jawi się jako alternatywa dla klasycznego zbioru MNIST. Posiada on dokładnie te same parametry co jego protoplasta. Zawiera on obrazy ubrań z dziesięciu różnych kategorii, co pozwala na przetestowanie modeli klasyfikacji obrazów na nieco bardziej złożonym zbiorze danych.

Dokonano normalizacji danych do zakresu  $[0, 1]$  na wartościach zmienoprzecinkowych, w miejsce domyślnego zakresu  $[0, 255]$  na wartościach całkowitych. Dla każdego ze zbiorów przygotowano dedykowany sekwencyjny model neuronowy. Oba modele zostały wstępnie przetrenowane oraz zapisane, aby możliwe było powtarzalne odtwarzanie badań. Dla przejrzystości, w dalszej części pracy zbiór danych MNIST będzie określany jako *Classic MNIST*, podczas gdy nazwa *Fashion MNIST* będzie używana bez zmian.

#### 3.3.1. Model podstawowy *DNN*

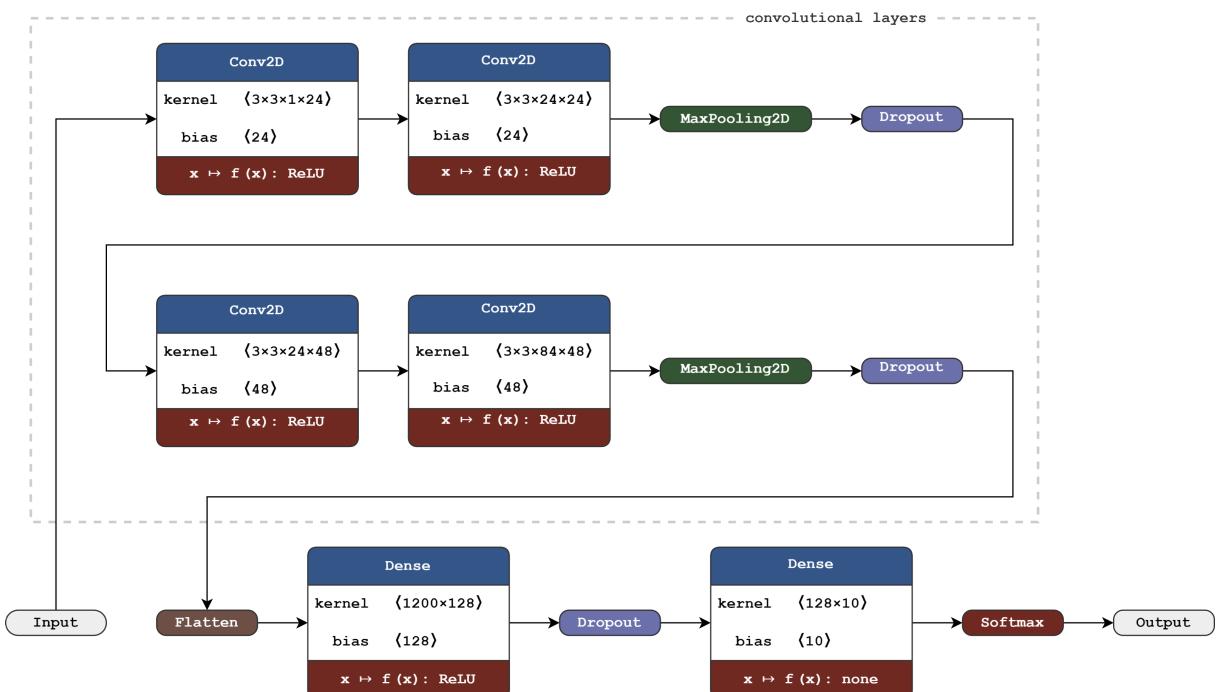
Przy projektowaniu modelu do klasyfikacji obrazów ze zbioru danych MNIST, zastosowano stosunkowo prostą architekturę sekwencyjną. Model ten składa się z jednej warstwy ukrytej (*Dense*), połączonej z warstwą spadkową (*Dropout*). Stanowi to popularną technikę regularyzacji, dzięki czemu unika się zjawiska *przeuczenia* modelu. Pozwoliło to na osiągnięcie 98.55% dokładności klasyfikacji dla podzbioru treningowego ( $p$ ) oraz 97.48% dokładności klasyfikacji dla podzbioru testowego ( $q$ ). Wyniki te są zbliżone do rezultatów osiąganych przez ludzi, co świadczy o skuteczności modelu w rozpoznawaniu cyfr. Model ten jest rozwinięciem modelu przedstawionego w dokumentacji TensorFlow [21].



Rys. 3.1: Architektura modelu podstawowego [opracowanie własne]

### 3.3.2. Model konwolucyjny CNN

Ze względu na większą złożoność zbioru danych Fashion MNIST, konieczne było zastosowanie bardziej zaawansowanej architektury modelu w porównaniu z modelem podstawowym, wykorzystanym do klasyfikacji cyfr ze zbioru MNIST. Zastosowano dwie podwójne warstwy konwolucyjne (*Conv2D*), wsparcie o warstwy redukcyjne (*MaxPooling2D*) i spadkowe (*Dropout*). Warstwy te pomagają zmniejszyć rozmiar map cech i zwiększyć poziom regularyzacji. Te rozszerzenia architektury umożliwiły osiągnięcie satysfakcjonujących wyników — dokładność klasyfikacji dla podzbioru treningowego ( $p$ ) wyniosła 95.38%, a dla podzbioru testowego ( $q$ ) 92.19%. Wyniki te ponownie są zbliżone do rezultatów osiąganych przez ludzi, co potwierdzają skuteczność zastosowanej architektury. Model ten jest rozwinięciem bazowego modelu przedstawionego w dokumentacji TensorFlow [22], który został zastosowany do klasyfikacji obrazów na zbiorze danych **CIFAR10**. W związku z tym należy uznać, iż architektura modelu jest wystarczająco rozbudowana, aby efektywnie radzić sobie z zadaniami na zbiorze Fashion MNIST.



Rys. 3.2: Architektura modelu rozszerzonego [opracowanie własne]

### 3.3.3. Modele rozszerzone

Na bazie podstawowych modeli DNN oraz CNN skonstruowano nowe, które zbiorczo określono mianem *rozszerzonych*. W modelach tych warstwy spadkowe **Dropout** zastąpiono **Warstwami szumu generatywnego**: szumem białym o rozkładzie jednostajnym, białym o rozkładzie normalnym oraz szumem Cauchy'ego. Warstwy te generowały 100% zadanego poziomu szumu podczas treningu, oraz 25% zadanego poziomu szumu podczas normalnej pracy. Parametry generacji dobrano w taki sposób, aby zapewnić zbliżony poziom dokładności klasyfikacji dla podzbioru treningowego ( $p$ ) oraz dla podzbioru testowego ( $q$ ), z tolerancją  $\pm 2\text{ p.p.}$ .

Taka konstrukcja modelu powinna zapewnić podobny poziom regularyzacji, co bazowe warstwy Dropout podczas treningu. Dodatkowo wprowadzenie warstw szumu generatywnego do architektury modelu wprowadza kontrolowaną losowość podczas normalnej pracy, co z kolei powinno utrudniać przeprowadzenie skutecznego ataku kontradydiktoryjnego. Uzyskane eksperymentalnie wyniki zawarto zbiorczo w tabelach poniżej:

Classic MNIST		
model	$p$ accuracy	$q$ accuracy
DNN	99.38%	97.90%
DNN + Cauchy	98.83%	97.18%
DNN + normal	99.22%	97.19%
DNN + uniform	99.02%	96.83%

Tab. 3.1: Porównanie dokładności klasyfikacji dla modeli dla modeli DNN

Fashion MNIST		
model	$p$ accuracy	$q$ accuracy
CNN	94.60%	91.64%
CNN + Cauchy	95.38%	92.34%
CNN + normal	94.10%	91.26%
CNN + uniform	95.25%	91.99%

Tab. 3.2: Porównanie dokładności klasyfikacji dla modeli dla modeli CNN

## 3.4. Zastosowane metryki

W związku z tym, iż podzbiory walidacyjne, na których prowadzono właściwe badania, były duże (10 000 elementów) i zrównoważone, to jako główną miarę wybrano dokładność klasyfikacji (ang. *accuracy*), a więc stosunek liczby poprawnych predykcji do ogólnej liczby próbek w zbiorze danych. Jest to jedna z najbardziej podstawowych miar jakościowych w dziedzinie klasyfikacji obrazów. Miarę tę wyznaczano dla każdego modelu podczas klasyfikacji danych zakłóconych w porównaniu do danych niezakłóconych.

Oprócz dokładności wyznaczano także macierze błędu, aby zwizualizować zmiany zachodzące w predykcji poszczególnych klas pod wpływem zwiększających się zakłóceń. Obliczano także energię  $\epsilon$  perturbacji, czyli sumę kwadratów różnic wszystkich pikseli pomiędzy obrazem oryginalnym a zakłóconym. Wszystkie badania zostały przeprowadzane na zbiorach testowych ( $q$ ), właściwych dla każdego z modeli neuronowych.

### 3.4.1. Dokładność klasyfikacji

Dokładność jest jedną z podstawowych metryk stosowanych do oceny wydajności modeli neuronowych, szczególnie w kontekście problemów klasyfikacyjnych. Definiuje się ją jako stosunek liczby poprawnie sklasyfikowanych próbek do całkowitej liczby prób w zbiorze testowym. Jest to miara, która odzwierciedla ogólną skuteczność modelu w przewidywaniu poprawnych etykiet. Wysoka dokładność wskazuje na dobrą wydajność modelu, lecz w przypadku niezrównoważonych zbiorów danych może być ona myląca. Formalnie wyraża się ją wzorem:

$$\text{accuracy} = \frac{Q_p}{Q_p + Q_n}$$

gdzie:

$Q_p$  – poprawne klasyfikacje (ang. *correct predictions*)

$Q_n$  – niepoprawne klasyfikacje (ang. *incorrect predictions*)

### 3.4.2. Energia perturbacji

Energia perturbacji odnosi się do miary, która ocenia istotność zmian w wektorze wejściowym. Definiowana jest jako suma kwadratów modyfikacji wprowadzonych do wektora wejściowego. Ta metryka pozwala na ilościową ocenę skali tych modyfikacji, czyli ocenia, jakauważalne są te zmiany, na przykład dla ludzkiego oka. Im wyższa wartość energii perturbacji, tym bardziejauważalne są wprowadzone zmiany. Matematycznie, energia perturbacji  $E$  może być wyrażona jako:

$$E = \sum_{i=1}^n (X_i - X'_i)^2$$

gdzie:

$n$  – liczba pikseli na obrazie

$X$  – wektor pikseli obrazu oryginalnego

$X'$  – wektor pikseli obrazu z perturbacjami

### 3.4.3. Macierz błędu

Macierz błędu jest narzędziem stosowanym do zbiorczej wizualizacji wydajności modelu klasyfikacyjnego poprzez porównanie jego przewidywanych etykiet z prawdziwymi etykietami. Umożliwia analizę błędów klasystycznych z perspektywy całego modelu, a nie dla wybranych przypadków. Konstruuje się ją jako kwadratową macierz, w której elementy  $M_{i,j}$  oznaczają liczbę przypadków rzeczywistej klasy  $i$  sklasyfikowanych jako klasa  $j$ . Typowa macierz błędu dla problemu z  $K$  klasami przedstawia się następująco:

$$\begin{bmatrix} M_{1,1} & M_{1,2} & \cdots & M_{1,K} \\ M_{2,1} & M_{2,2} & \cdots & M_{2,K} \\ \vdots & \vdots & \ddots & \vdots \\ M_{K,1} & M_{K,2} & \cdots & M_{K,K} \end{bmatrix}$$

gdzie:

$i$  – rzeczywista klasa obrazu

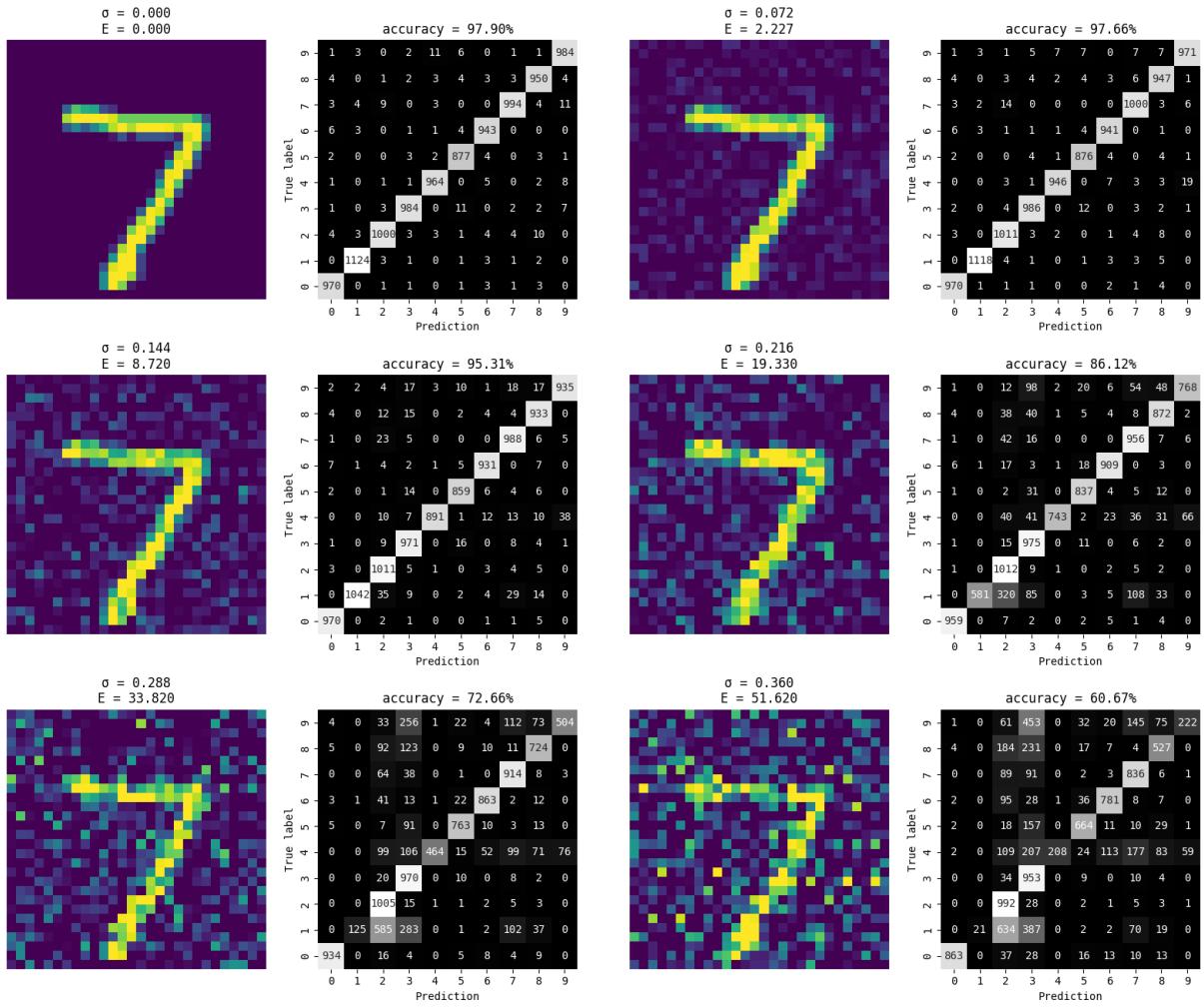
$j$  – predykcja klasy dla obrazu

$K$  – liczba klas w zbiorze danych

$M$  – liczba wystąpień etykiety

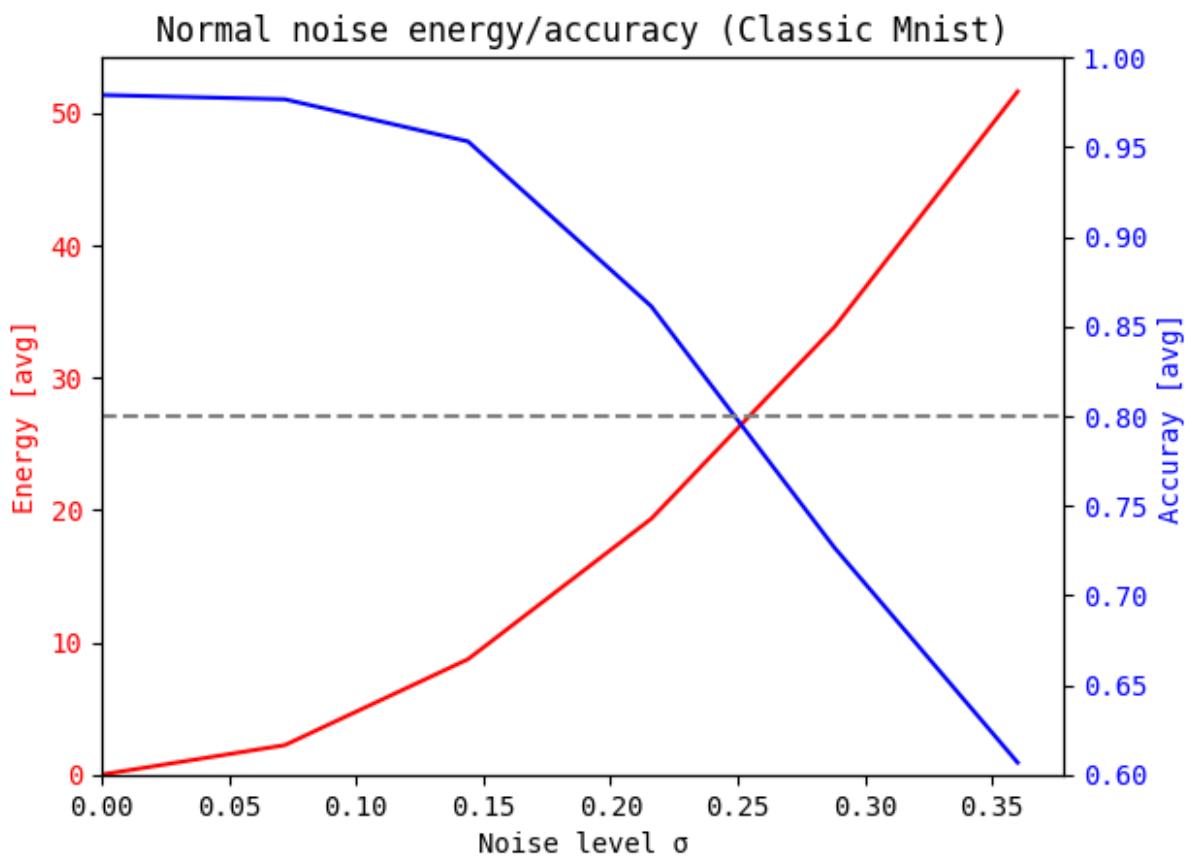
## 3.5. Standardowy zestaw badawczy

Aby przeprowadzić dokładną i sprawiedliwą analizę wyników, opracowano tzw. standardowy zestaw badawczy. Składa się on z sześciu iteracji, z których każda zwraca cztery kluczowe elementy oceny. Po pierwsze, dla każdej iteracji obliczana jest uśredniona dokładność klasyfikacji, która pozwala ocenić ogólną skuteczność modelu. Po drugie, mierzy się uśredzoną energię perturbacji, co umożliwia oszacowanie, jak duże były zmiany wywołane zakłóceniami. Trzecim elementem jest macierz pomyłek, która dostarcza szczegółowych informacji na temat klasyfikacji, wskazując liczbę prawidłowych i błędnych przypisań do poszczególnych klas. Analiza macierzy pomyłek pozwala na bardziej szczegółowe zrozumienie, gdzie model popełnia błędy, czy to w nadmiarowej klasyfikacji jednych klas kosztem innych, czy w niedokładności przypisywania. Ostatnim składnikiem jest podgląd pierwszego obrazu z podzbioru danych, na który nałożono zakłócenia, co pozwala wizualnie ocenić wpływ perturbacji na dane wejściowe. Wizualizacja ta jest istotna, ponieważ pozwala na bezpośrednią ocenę efektów zakłóceń na oryginalne obrazy, a także na porównanie różnych rodzajów zakłóceń z różnych prób badawczych. Taki kompleksowy zestaw danych umożliwia wieloaspektową i rzetelną ocenę wydajności badanych modeli. Współne zastosowanie tych czterech elementów zapewnia pełny obraz działania modelu, od ogólnej skuteczności, przez podatność na zakłócenia, po szczegółowe informacje o błędach klasyfikacji i wizualną ocenę wpływu zakłóceń.



Rys. 3.3: Wygląd przykładowego standardowego zestawu badawczego [opracowanie własne]

Równolegle ze standardowym zestawem badawczym, przygotowano także wykresy zależności energii perturbacji oraz dokładności klasyfikacji od parametrów zakłóceń. Wykresy te służą do kompaktowej reprezentacji danych z danej próby badawczej, a także do lepszego zwizualizowania ogólnych trendów. Taka prezentacja wyników ułatwia wizualne zrozumienie zależności między badanymi parametrami, co jest szczególnie przydatne w analizach porównawczych różnych zakłóceń. Wykresy te przygotowano na podstawie przebiegów z kolejnych prób badawczych dla różnych parametrów danego zakłócenia. Analiza ta pozwala na precyzyjne przedstawienie, jak zmiany w parametrach zakłóceń wpływają na energię perturbacji oraz na dokładność klasyfikacji modelu. Dzięki temu możliwe jest zidentyfikowanie specyficznych punktów krytycznych, w których zakłócenia zaczynają znaczco obniżać wydajność modelu. Ponadto, na podstawie wykresów przedstawiających zależność między energią perturbacji a dokładnością klasyfikacji możliwa jest identyfikacja optymalnych parametrów zakłóceń, które sprzyjają pożądanemu zachowaniu modelu. Umożliwia to opracowanie bardziej złożonych scenariuszy badawczych, w których istotna jest powtarzalność uzyskiwanych wyników.



Rys. 3.4: Wygląd przykładowego wykresu zależności próby badawczej. Na czerwono: zależności energii perturbacji od wartości zmiennej sterującej zakłóceniem; na niebiesko: zależność dokładności klasyfikacji od wartości zmiennej sterującej zakłóceniem [opracowanie własne]

## Rozdział 4

# Badania symulacyjne

W tym rozdziale fazie przeprowadzono testy jakościowe każdego z wybranych modeli w środowisku zakłóconym. Przedstawione zostaną wyniki badań symulacyjnych dotyczących wpływu zakłóceń na modele neuronowe oraz skuteczności zaproponowanych strategii regularyzacji. Celem jest analiza, w jaki sposób różne rodzaje losowych fluktuacji oddziałują na stabilność modeli neuronowych, jak zniekształcają one oryginalne dane oraz czy możliwe jest wykorzystanie zakłóceń stochastycznych do wzmacnienia obrony przed zakłóceniami kontradyktoryjnymi.

### 4.1. Wpływ zakłóceń stochastycznych na proces klasyfikacji obrazów

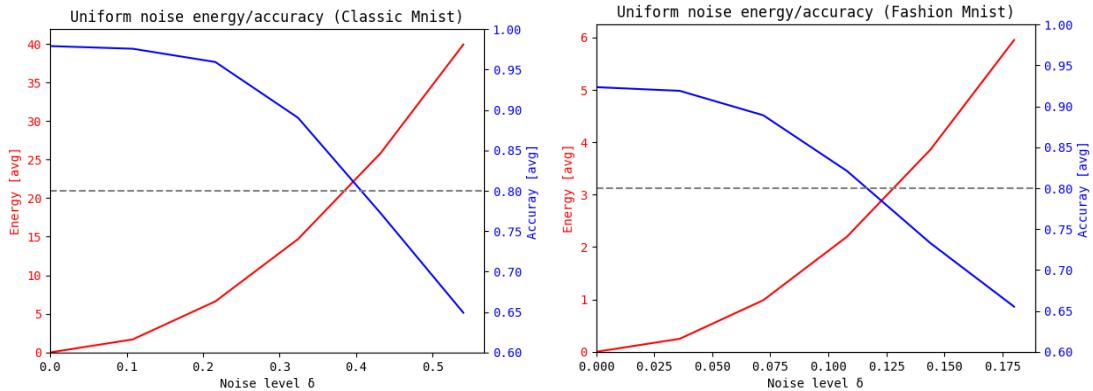
Badania przeprowadzono w trzech seriach: szum biały o rozkładzie jednostajnym, szum biały o rozkładzie Normalnym, szum Cauchy'ego. Każda z tych serii miała na celu zbadanie wpływu tych typów zakłóceń na dokładność klasyfikacji. Dla każdego rodzaju zakłóceń przeprowadzono eksperymenty z pięcioma różnymi parametrami sterującymi, specyficznymi dla każdego typu szumu, które wzrastały liniowo. Każda seria obejmowała również próbę kontrolną, w której obrazy nie były poddane żadnym zakłóceniom. Dzięki temu możliwe było porównanie wyników i wyciągnięcie wniosków na temat wpływu zakłóceń na model.

Parametry zakłóceń zostały dobrane w taki sposób, aby w ostatniej próbie dokładność klasyfikacji wynosiła około 65% ( $\pm 5\text{ p.p.}$ ) na zbiorze walidacyjnym. W każdej serii zwiększano intensywność zakłóceń liniowo. Badania porównawcze przeprowadzono dla dokładności wynoszącej równo 80%. Poniżej tego progu skuteczność modelu jest znacznie ograniczona, co sprawia, że jego użyteczność w rzeczywistych zadaniach klasyfikacyjnych staje się wysoce wątpliwa. W celu zapewnienia spójności i powtarzalności wyników każde doświadczenie zostało przeprowadzone wielokrotnie, a uzyskane dane zostały uśrednione. Procedura ta pozwoliła na zminimalizowanie wpływu przypadkowych błędów i zapewniła wiarygodność wyników.

### 4.1.1. Wpływ szumu białego o rozkładzie jednostajnym

Parametrem sterującym szumu była wartość graniczna  $\delta$ , która wyznaczała zakres  $\xi \in [-\delta, +\delta]$ . Badania przeprowadzono dla zakłóceń z zakresu  $\delta \in [0.00, 0.54]$  dla zbioru Classic MNIST, oraz  $\delta \in [0.00, 0.18]$  dla zbioru Fashion MNIST. Funkcja gęstości prawdopodobieństwa badanego szumu opisana jest wzorem:

$$f(x) = \frac{1}{2 * \delta} \quad \text{gdzie: } \delta > 0 \quad \text{oraz: } x \in [-\delta, +\delta]$$



Rys. 4.1: Wykresy zależności energii perturbacji oraz dokładności klasyfikacji dla próby badawczej.

Energy at 80% accuracy		
dataset	avg. energy	regularization
Classic MNIST (DNN)	$\approx 2.3 \cdot 10^1$	Dropout
Fashion MNIST (CNN)	$\approx 2.6 \cdot 10^0$	Dropout

Tab. 4.1: Tabela zależności energii perturbacji oraz dokładności klasyfikacji dla próby badawczej

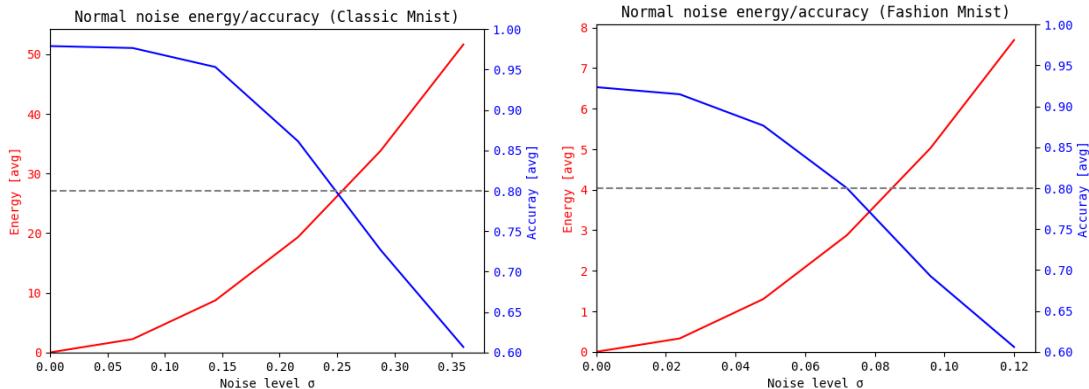
### Wynik badania

Model DNN operujący na zbiorze Classic MNIST wymagał około 9 razy większej energii perturbacji, do osiągnięcia zadanego punktu pracy 80% dokładności klasyfikacji, w stosunku do modelu CNN operującego na zbiorze Fashion MNIST. Dla obu modeli zaobserwowano efekt ponadliniowego wzrostu energii perturbacji w odpowiedzi na liniowo wzrastający parametr  $\delta$  sterujący zakłóceniem. Równocześnie stwierdzono, że dokładność klasyfikacji malała ponadliniowo w miarę liniowego wzrostu tego samego parametru. Energia perturbacji okazała się być negatywnie skorelowana z dokładnością klasyfikacji, co oznacza, że ogólnie wyższe wartości energii perturbacji prowadziły do niższej dokładności klasyfikacji.

### 4.1.2. Wpływ szumu białego o rozkładzie Normalnym

Parametrem sterującym szumu było odchylenie standardowe  $\sigma$  od wartości oczekiwanej równej  $\mu = 0$ . Badania przeprowadzono dla zakłóceń z zakresu  $\sigma \in [0.00, 0.36]$  dla zbioru Classic MNIST, oraz  $\sigma \in [0.00, 0.12]$  dla zbioru Fashion MNIST. Funkcja gęstości prawdopodobieństwa badanego szumu opisana jest wzorem:

$$f(x) = \frac{1}{\sigma \cdot \sqrt{2\pi}} \cdot e^{-\frac{x^2}{2\sigma^2}} \quad \text{gdzie: } \sigma > 0 \quad \text{oraz: } x \in \mathbb{R}$$



Rys. 4.2: Wykresy zależności energii perturbacji oraz dokładności klasyfikacji dla próby badawczej

Energy at 80% accuracy		
dataset	avg. energy	regularization
Classic MNIST (DNN)	$\approx 2.6 \cdot 10^1$	Dropout
Fashion MNIST (CNN)	$\approx 2.8 \cdot 10^0$	Dropout

Tab. 4.2: Tabela zależności energii perturbacji oraz dokładności klasyfikacji dla próby badawczej

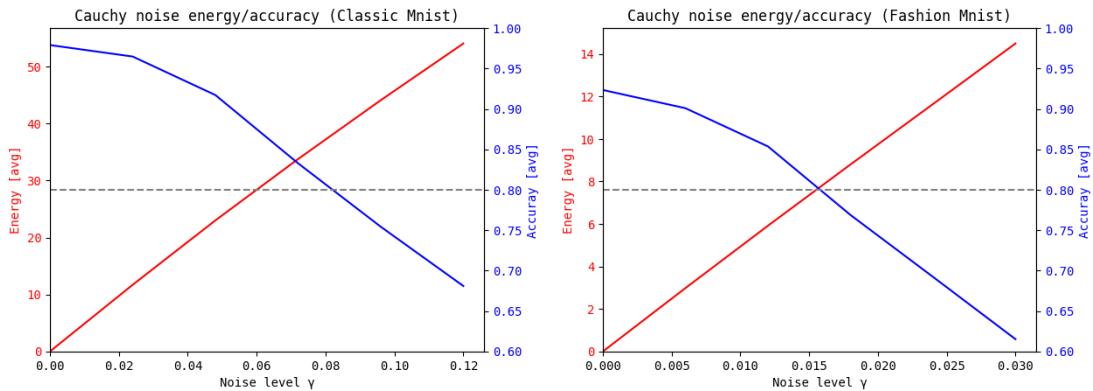
### Wynik badania

Model DNN operujący na zbiorze Classic MNIST wymagał około 9 razy większej energii perturbacji, do osiągnięcia zadanego punktu pracy 80% dokładności klasyfikacji, w stosunku do modelu CNN operującego na zbiorze Fashion MNIST. Dla obu modeli zaobserwowano efekt ponadliniowego wzrostu energii perturbacji w odpowiedzi na liniowo wzrastający parametr  $\sigma$  sterujący zakłóceniem. Równocześnie stwierdzono, że dokładność klasyfikacji malała ponadliniowo w miarę liniowego wzrostu tego samego parametru. Energia perturbacji okazała się być negatywnie skorelowana z dokładnością klasyfikacji, co oznacza, że ogólnie wyższe wartości energii perturbacji prowadziły do niższej dokładności klasyfikacji. Ogólny poziom energii potrzebny do osiągnięcia zadanego punktu pracy wzrósł odpowiednio o  $\approx 13\%$  i  $\approx 7.5\%$ , względem próby z szumem białym o rozkładzie jednostajnym.

### 4.1.3. Wpływ szumu Cauchy'ego

Parametrem sterującym szumu było wzmacnienie  $\gamma$ . Sam szum modelowany był jako iloraz dwóch zmiennych losowych z rozkładu Normalnego o wartości oczekiwanej równej  $\mu = 0$  i odchyleniu standardowym  $\sigma = 1$ , pomnożony przez wartość wzmacnienia. Badania przeprowadzono dla zakłóceń z zakresu  $\gamma \in [0.00, 0.12]$  dla zbioru Classic MNIST, oraz  $\gamma \in [0.00, 0.03]$  dla zbioru Fashion MNIST. Funkcja gęstości prawdopodobieństwa badanego szumu opisana jest wzorem:

$$f(x) = \frac{1}{\pi\gamma \cdot [1 + (\frac{x}{\gamma})^2]} \quad \text{gdzie: } \gamma > 0 \quad \text{oraz: } x \in \mathbb{R}$$



Rys. 4.3: Tabela zależności energii perturbacji oraz dokładności klasyfikacji dla próby badawczej

Energy at 80% accuracy		
dataset	avg. energy	regularization
Classic MNIST (DNN)	$\approx 4.8 \cdot 10^1$	Dropout
Fashion MNIST (CNN)	$\approx 7.7 \cdot 10^0$	Dropout

Tab. 4.3: Wykresy zależności energii perturbacji oraz dokładności klasyfikacji dla próby badawczej

### Wynik badania

Model DNN operujący na zbiorze Classic MNIST wymagał około 6 razy większej energii perturbacji, do osiągnięcia zadanego punktu pracy 80% dokładności klasyfikacji, w stosunku do modelu CNN operującego na zbiorze Fashion MNIST. Dla obu modeli zaobserwowano efekt liniowego wzrostu energii perturbacji w odpowiedzi na liniowo wzrastający parametr  $\gamma$  sterujący zakłóceniem. Równocześnie stwierdzono, że dokładność klasyfikacji malała ponadliniowo w miarę liniowego wzrostu tego samego parametru. Energia perturbacji okazała się być negatywnie skorelowana z dokładnością klasyfikacji, co oznacza, że ogólnie wyższe wartości energii perturbacji prowadziły do niższej dokładności klasyfikacji. Ogólny poziom energii potrzebny do osiągnięcia zadanego punktu pracy wzrósł odpowiednio o  $\approx 108\%$  i  $\approx 196\%$ , względem próby z szumem białym o rozkładzie jednostajnym.

#### **4.1.4. Konkluzja wyników analizy symulacyjnej**

Badania wykazały, że modele DNN i CNN wykazują różną wrażliwość na zakłócenia w postaci szumu białego o rozkładzie jednostajnym, normalnym oraz szumu Cauchy'ego. Model DNN operujący na zbiorze Classic MNIST wymagał znacznie większej energii perturbacji do osiągnięcia pożdanego spadku dokładności klasyfikacji w porównaniu do modelu CNN na zbiorze Fashion MNIST. Najmniejszą różnicę ( $\approx 6 \times$ ) zaobserwowano w przypadku szumu Cauchy'ego. W wypadku szumów białych różnice te były na podobnym poziomie ( $\approx 9 \times$ ). Dla szumu Cauchy'ego wzrost energii perturbacji był liniowo skorelowany ze wzrostem parametru sterującego zakłóceniem, dla szumów białych korelacja ta była ponadlinowa.

Ponadto, dla obu modeli zaobserwowano ponadliniowy spadek dokładności klasyfikacji w odpowiedzi na liniowy wzrost parametrów zakłócenia, niezależnie od rodzaju szumu. Wzrost energii perturbacji był negatywnie skorelowany z dokładnością klasyfikacji, co oznacza, że wyższe wartości energii prowadziły do niższej dokładności. W przypadku szumu normalnego poziom energii potrzebnej do osiągnięcia zadanego punktu pracy wzrósł odpowiednio o  $\approx 13\%$  i  $\approx 7.5\%$  w porównaniu do szumu białego o rozkładzie jednostajnym. Natomiast szum Cauchy'ego wymagał jeszcze wyższych wartości energii, zwiększając je o  $\approx 108\%$  i  $\approx 196\%$  odpowiednio dla DNN i CNN. Implikuje to, że największą destruktywnością cechuje się szum biały o rozkładzie jednostajnym.

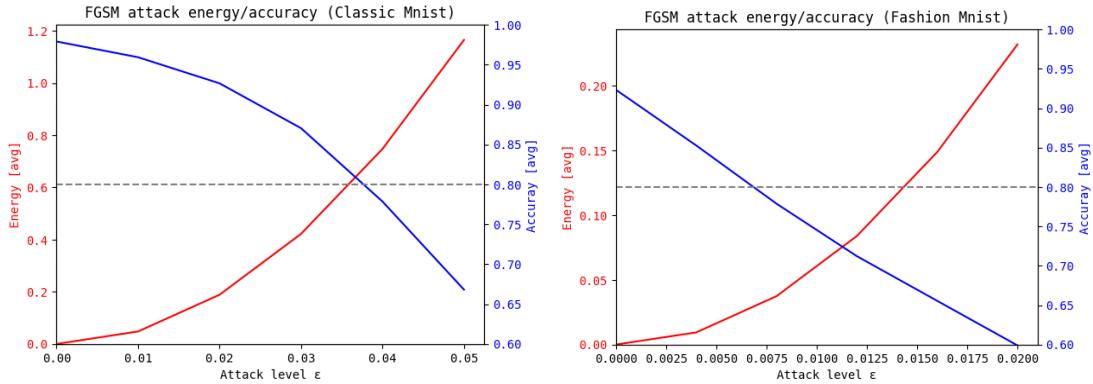
## **4.2. Wpływ zakłóceń kontradydaktryjnych na proces klasyfikacji obrazów**

Badania przeprowadzono w trzech seriach: atak FGSM, atak LSGSM, atak DeepFool. Każda z tych serii miała na celu zbadanie wpływu tychże typów zakłóceń na dokładność klasyfikacji. Dla każdego rodzaju zakłóceń przeprowadzono eksperymenty z pięcioma różnymi wartościami parametru sterującego  $\epsilon$ , który wzrastał liniowo. Każda seria obejmowała również próbę kontrolną, w której obrazy nie były poddane żadnym zakłóceniom. Dzięki temu możliwe było porównanie wyników i wyciągnięcie wniosków na temat wpływu zakłóceń na model.

Parametry zakłóceń zostały dobrane w taki sposób, aby w ostatniej próbie dokładność klasyfikacji wynosiła około 65% ( $\pm 5\text{ p.p.}$ ) na zbiorze walidacyjnym. W każdej serii zwiększano intensywność zakłóceń liniowo. Badania porównawcze przeprowadzono dla dokładności wynoszącej równo 80%. Poniżej tego progu skuteczność modelu jest znacznie ograniczona, co sprawia, że jego użyteczność w rzeczywistych zadaniach klasyfikacyjnych staje się wysoce wątpliwa. W celu zapewnienia spójności i powtarzalności wyników każde doświadczenie zostało przeprowadzone wielokrotnie, a uzyskane dane zostały uśrednione. Procedura ta pozwoliła na zminimalizowanie wpływu przypadkowych błędów i zapewniła wiarygodność wyników.

### 4.2.1. Wpływ zakłóceń ataku FGSM

Parametrem sterującym zakłócenia była wartość  $\epsilon$ , która wyznaczała wzmocnienie kierunku gradientu funkcji straty. Badania przeprowadzono dla zakłóceń z zakresu  $\epsilon \in [0.00, 0.05]$  dla zbioru Classic MNIST, oraz  $\epsilon \in [0.00, 0.02]$  dla zbioru Fashion MNIST. Wartości parametru  $\epsilon$  są tożsame jak w wypadku algorytmu LESGSM.



Rys. 4.4: Wykresy zależności energii perturbacji oraz dokładności klasyfikacji dla próby badawczej

Energy at 80% accuracy		
dataset	avg. energy	regularization
Classic MNIST (DNN)	$\approx 6.6 \cdot 10^{-1}$	Dropout
Fashion MNIST (CNN)	$\approx 3.1 \cdot 10^{-2}$	Dropout

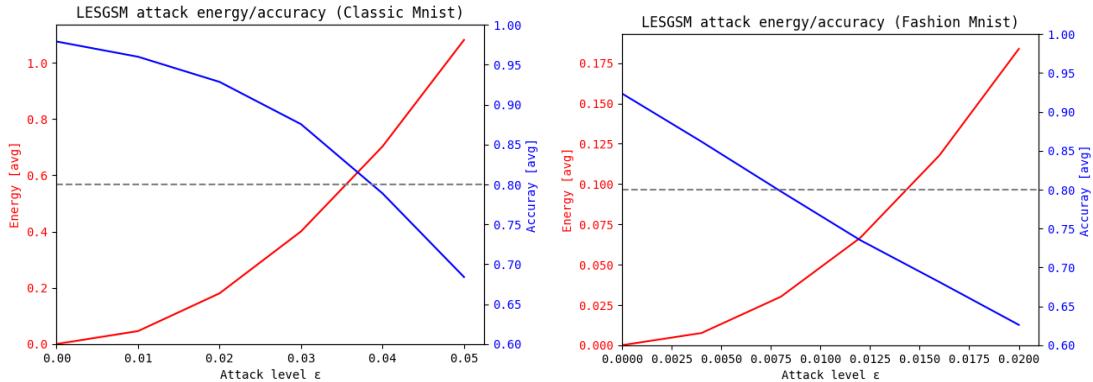
Tab. 4.4: Tabela zależności energii perturbacji oraz dokładności klasyfikacji dla próby badawczej

### Wynik badania

Model DNN operujący na zbiorze Classic MNIST wymagał około 21 razy większej energii perturbacji, do osiągnięcia zadanego punktu pracy 80% dokładności klasyfikacji, w stosunku do modelu CNN operującego na zbiorze Fashion MNIST. Dla obu modeli zaobserwowano efekt liniowego wzrostu energii perturbacji w odpowiedzi na liniowo wzrastający parametr  $\epsilon$  sterujący zakłóceniem. Równocześnie stwierdzono, że dokładność klasyfikacji malała ponadliniowo w miarę liniowego wzrostu tego samego parametru dla modelu DNN, lecz w tempie nieznacznie wolniejszym niż liniowy dla modelu CNN. Energia perturbacji okazała się być negatywnie skorelowana z dokładnością klasyfikacji, co oznacza, że ogólnie wyższe wartości energii perturbacji prowadziły do niższej dokładności klasyfikacji.

## 4.2.2. Wpływ zakłóceń ataku *LESGSM*

Parametrem sterującym zakłócenia była wartość  $\epsilon$ , która wyznaczała bazę wzmacnienia kierunku gradientu funkcji straty. Badania przeprowadzono dla zakłóceń z zakresu  $\epsilon \in [0.00, 0.05]$  dla zbioru Classic MNIST, oraz  $\epsilon \in [0.00, 0.02]$  dla zbioru Fashion MNIST. Wartości parametru  $\epsilon$  są tożsame jak w wypadku algorytmu FGSM.



Rys. 4.5: Wykresy zależności energii perturbacji oraz dokładności klasyfikacji dla próby badawczej

Energy at 80% accuracy		
dataset	avg. energy	regularization
Classic MNIST (DNN)	$\approx 6.5 \cdot 10^{-1}$	Dropout
Fashion MNIST (CNN)	$\approx 2.9 \cdot 10^{-2}$	Dropout

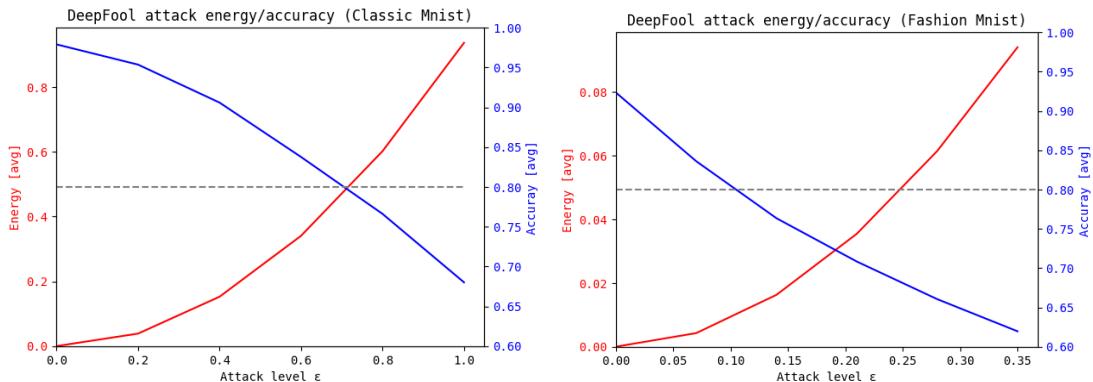
Tab. 4.5: Tabela zależności energii perturbacji oraz dokładności klasyfikacji dla próby badawczej

## Wynik badania

Model DNN operujący na zbiorze Classic MNIST wymagał około 22 razy większej energii perturbacji, do osiągnięcia zadanego punktu pracy 80% dokładności klasyfikacji, w stosunku do modelu CNN operującego na zbiorze Fashion MNIST. Dla obu modeli zaobserwowano efekt liniowego wzrostu energii perturbacji w odpowiedzi na liniowo wzrastający parametr  $\epsilon$  sterujący zakłóceniem. Równocześnie stwierdzono, że dokładność klasyfikacji malała ponadliniowo w miarę liniowego wzrostu tego samego parametru dla modelu DNN, lecz w tempie nieznacznie wolniejszym niż liniowy dla modelu CNN. Energia perturbacji okazała się być negatywnie skorelowana z dokładnością klasyfikacji, co oznacza, że ogólnie wyższe wartości energii perturbacji prowadziły do niższej dokładności klasyfikacji. Ogólny poziom energii potrzebny do osiągnięcia zadanego punktu pracy spadł odpowiednio o  $\approx 1.5\%$  i  $\approx 6.5\%$ , względem próby z szumem białym o rozkładzie jednostajnym.

### 4.2.3. Wpływ zakłóceń ataku *DeepFool*

Parametrem sterującym zakłócenia była wartość  $\epsilon$ , która wyznaczała bazę wzmacnienia kierunku gradientu funkcji straty. Badania przeprowadzono dla zakłóceń z zakresu  $\epsilon \in [0.00, 1.00]$  dla zbioru Classic MNIST, oraz  $\epsilon \in [0.00, 0.35]$  dla zbioru Fashion MNIST. Wartości parametru  $\epsilon$  różnią się od względem algorytmów FGSM i LESGSM, lecz zostały dobrane w sposób zapewniający osiągnięcie podobnych parametrów.



Rys. 4.6: Wykresy zależności energii perturbacji oraz dokładności klasyfikacji dla próby badawczej

Energy at 80% accuracy		
dataset	avg. energy	regularization
Classic MNIST (DNN)	$\approx 4.9 \cdot 10^{-1}$	Dropout
Fashion MNIST (CNN)	$\approx 1.0 \cdot 10^{-2}$	Dropout

Tab. 4.6: Tabela zależności energii perturbacji oraz dokładności klasyfikacji dla próby badawczej

## Wynik badania

Model DNN operujący na zbiorze Classic MNIST wymagał około 49 razy większej energii perturbacji, do osiągnięcia zadanego punktu pracy 80% dokładności klasyfikacji, w stosunku do modelu CNN operującego na zbiorze Fashion MNIST. Dla obu modeli zaobserwowano efekt liniowego wzrostu energii perturbacji w odpowiedzi na liniowo wzrastający parametr  $\epsilon$  sterujący zakłóceniem. Równocześnie stwierdzono, że dokładność klasyfikacji mała ponadliniowo w miarę liniowego wzrostu tego samego parametru dla modelu DNN, lecz w tempie nieznacznie wolniejszym niż liniowe dla modelu CNN. Energia perturbacji okazała się być negatywnie skorelowana z dokładnością klasyfikacji, co oznacza, że ogólnie wyższe wartości energii perturbacji prowadziły do niższej dokładności klasyfikacji. Ogólny poziom energii potrzebny do osiągnięcia zadanego punktu pracy spadł odpowiednio o  $\approx 25\%$  i  $\approx 67\%$ , względem próby z szumem białym o rozkładzie jednostajnym.

#### **4.2.4. Konkluzja wyników analizy symulacyjnej**

Badania wykazały, że modele DNN i CNN wykazują różną wrażliwość na zakłócenia kontradyktoryjne, lecz ogólnie wyższą względem zakłóceń stochastycznych. Różnica ta sięga dwóch rzędów wielkości dla obu badanych modeli. Model DNN operujący na zbiorze Classic MNIST wymagał znacznie większej energii perturbacji do osiągnięcia pożdanego spadku dokładności klasyfikacji w porównaniu do modelu CNN na zbiorze Fashion MNIST. Najmniejszą różnicę ( $\approx 21 \times$ ) zaobserwowano w przypadku ataku FGSM. Różnica ta była minimalnie większa dla ataku LESGSM ( $\approx 49 \times$ ) oraz istotnie większa dla ataku DeepFool ( $\approx 6 \times$ ). Dla każdego z badanych algorytmów kontradyktoryjnych wzrost energii perturbacji był ponadliniowo skorelowany ze wzrostem parametru  $\epsilon$  sterującego zakłóceniem.

Ponadto, dla modeli DNN zaobserwowano ponadliniowy spadek dokładności klasyfikacji w odpowiedzi na liniowy wzrost parametrów zakłócenia. Spadek ten był liniowy w dla zakłóceń generowanych przez algorytmy FGSM i LESGSM, a dla algorytmu DeepFool postępował w tempie nieznacznie wolniejszym niż liniowe. Wzrost energii perturbacji był negatywnie skorelowany z dokładnością klasyfikacji, co oznacza, że wyższe wartości energii prowadziły do niższej dokładności. W przypadku ataku LESGSM poziom energii potrzebnej do osiągnięcia zadanego punktu pracy spadł odpowiednio o  $\approx 1.5\%$  i  $\approx 6.5\%$  w porównaniu do bazowego algorytmu FGSM. Natomiast szum Cauchy'ego wymagał jeszcze niższych wartości energii, obniżając je o  $\approx 25\%$  i  $\approx 67\%$  odpowiednio dla DNN i CNN.

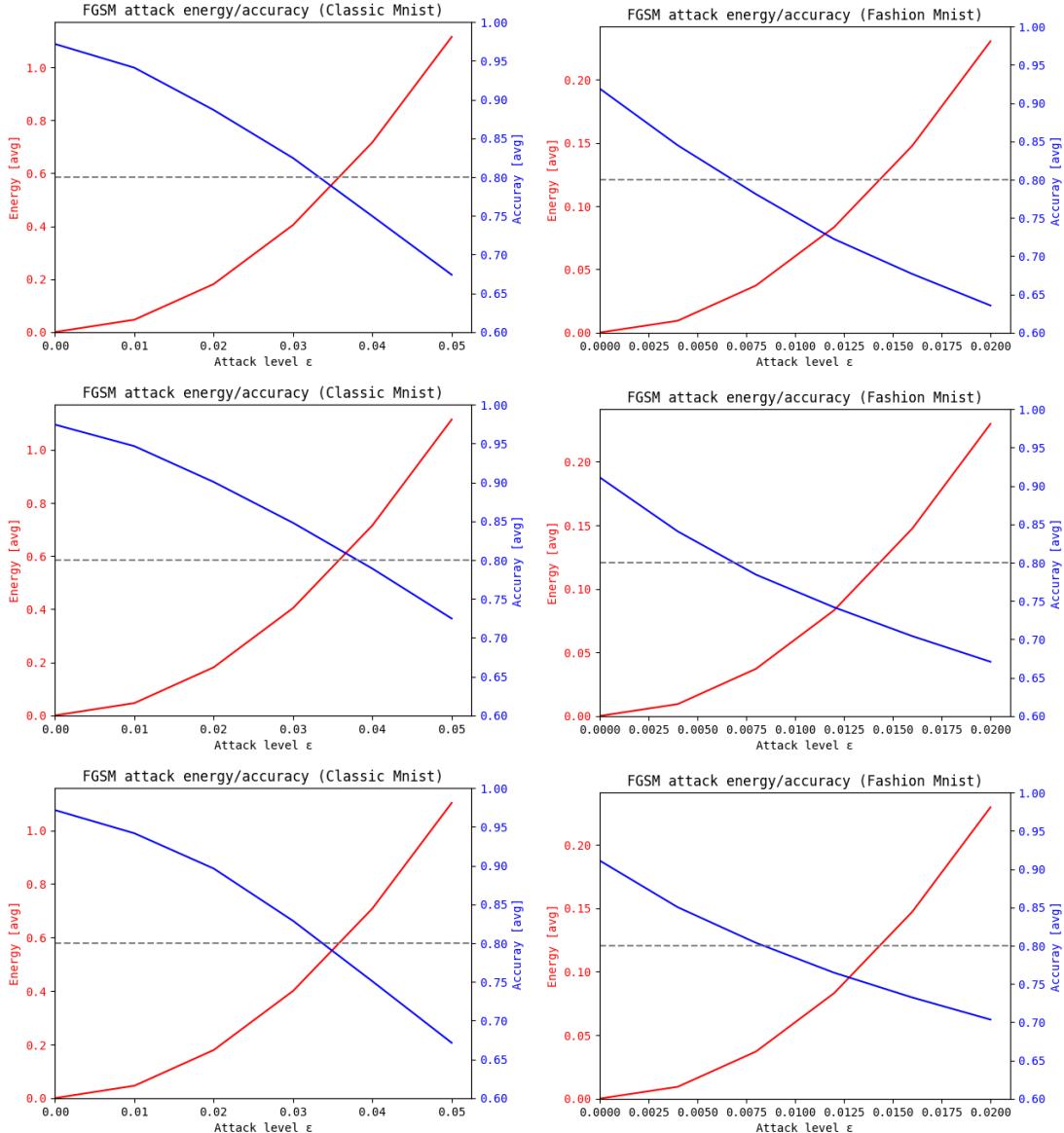
### **4.3. Ocena skuteczności strategii regularyzacji szumem**

Badania przeprowadzono w trzech seriach: atak FGSM, atak LESGSM, atak DeepFool na trzech różnych strategiach regularyzacji szumem, kolejno: szumem białym o rozkładzie jednostajnym, szumem białym o rozkładzie Normalnym, szumem Cauchy'ego. Każda z tych serii miała na celu zbadanie wpływu tychże typów zakłóceń na dokładność klasyfikacji. Dla każdego rodzaju zakłóceń przeprowadzono eksperymenty z pięcioma różnymi wartościami parametru sterującego  $\epsilon$ , tożsamymi, jak w badaniach wpływu zakłóceń kontradyktoryjnych na proces klasyfikacji obrazów w modelach podstawowych. Każda seria obejmowała również próbę kontrolną, w której obrazy nie były poddane żadnym zakłóceniom. Dzięki temu możliwe było porównanie wyników i wyciągnięcie wniosków na temat wpływu zakłóceń na model.

W każdej serii zwiększano intensywność zakłóceń liniowo. Badania porównawcze przeprowadzono dla dokładności wynoszącej równe 80% oraz dla ostatniej wartości parametru  $\epsilon$  w danej próbie badawczej. Dzięki temu możliwe jest nie tylko punktowe porównanie uzyskanych wyników względem prób kontrolnych, lecz wyznaczenie ogólnego trendu tychże zmian. W celu zapewnienia spójności i powtarzalności wyników każde doświadczenie zostało przeprowadzone wielokrotnie, a uzyskane dane zostały uśrednione. Procedura ta pozwoliła na zminimalizowanie wpływu przypadkowych błędów i zapewniła wiarygodność wyników.

### 4.3.1. Porównanie strategii regularyzacji szumem dla ataku *FGSM*

Parametrem sterującym zakłócenia była wartość  $\epsilon$ , która wyznaczała bazę wzmocnienia kierunku gradientu funkcji straty. Badania przeprowadzono dla zakłóceń z zakresu  $\epsilon \in [0.00, 0.05]$  dla zbioru Classic MNIST, oraz  $\epsilon \in [0.00, 0.02]$  dla zbioru Fashion MNIST.



Rys. 4.7: Porównanie wyników poszczególnych prób badawczych dla ataku FGSM

Energy at 80% accuracy			
dataset	avg. energy	$\Delta$ energy	regularization
Classic MNIST (DNN)	$\approx 5.2 \cdot 10^{-1}$	-21.2%	Uniform
Classic MNIST (DNN)	$\approx 6.6 \cdot 10^{-1}$	$\pm 0.00\%$	Normal
Classic MNIST (DNN)	$\approx 5.2 \cdot 10^{-1}$	-21.2%	Cauchy
Fashion MNIST (CNN)	$\approx 3.0 \cdot 10^{-2}$	-3.23%	Uniform
Fashion MNIST (CNN)	$\approx 3.1 \cdot 10^{-2}$	$\pm 0.00\%$	Normal
Fashion MNIST (CNN)	$\approx 4.1 \cdot 10^{-2}$	+32.3%	Cauchy

Tab. 4.7: Tabela zależności energii perturbacji oraz dokładności klasyfikacji dla serii badawczej

Accuracy at last $\epsilon$ parameter			
dataset	avg. accuracy	$\Delta$ accuracy	regularization
Classic MNIST (DNN)	$\approx 67\%$	$\pm 0.00\%$	Uniform
Classic MNIST (DNN)	$\approx 72\%$	$+7.46\%$	Normal
Classic MNIST (DNN)	$\approx 67\%$	$\pm 0.00\%$	Cauchy
Fashion MNIST (CNN)	$\approx 63\%$	$+5.00\%$	Uniform
Fashion MNIST (CNN)	$\approx 67\%$	$+11.6\%$	Normal
Fashion MNIST (CNN)	$\approx 70\%$	$+16.6\%$	Cauchy

Tab. 4.8: Tabela zależności dokładności klasyfikacji od ostatniego parametru  $\epsilon$  dla serii badawczej

## Wynik badania

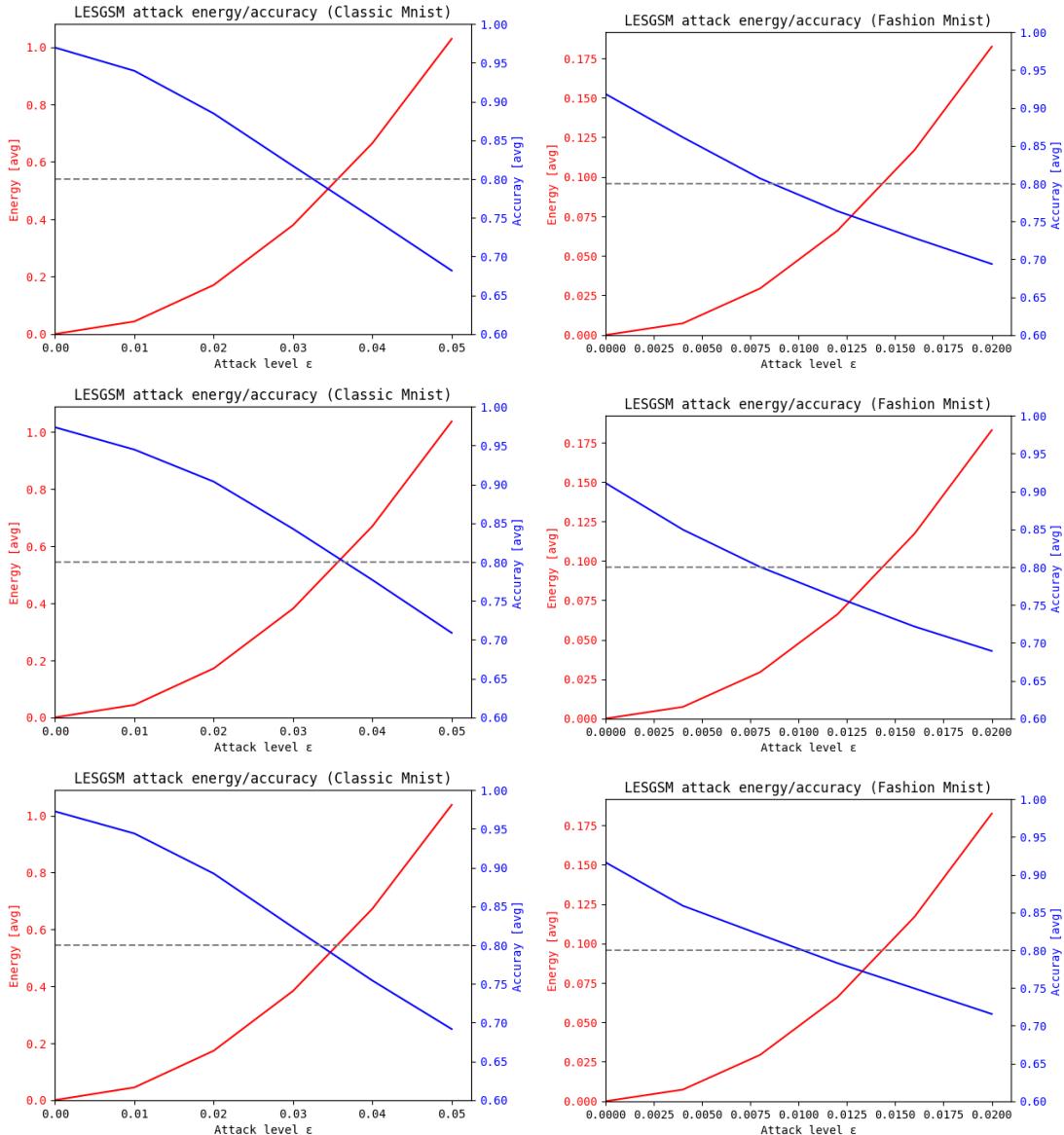
W punkcie pracy 80% dokładności klasyfikacji wykazano ogólną tendencję spadkową energii zarówno dla modelu DNN operującego na zbiorze Classic MNIST, jak i dla modelu CNN operującego na zbiorze Fashion MNIST. Zmiany te dotyczyły modeli wyposażonych w warstwy generatywne szumu białego o rozkładzie jednostajnym, oraz szumu Cauchy'ego. Zmiany te były na podobnym poziomie dla obu modeli. Wyjątkiem był model CNN wyposażony w warstwy generatywne szumu Cauchy'ego, dla którego stwierdzono istotny wzrost energii w zadanym punkcie pracy, a nie jej spadek. Natomiast w wypadku modeli (zarówno DNN jak i CNN) wyposażonych w warstwy generatywne szumu białego o rozkładzie normalnym nie zaobserwowano znaczących zmian energii perturbacji.

Końcowa dokładność klasyfikacji wszystkich modeli dla ostatniego badanego parametru  $\epsilon$  była na podobnym poziomie dla modeli DNN wyposażonych w warstwy generatywne szumu białego o rozkładzie jednostajnym oraz szumu Cauchy'ego. Zaobserwowano niewielki spadek skuteczności ataku dla modelu wyposażonego w warstwy generatywne szumu białego o rozkładzie normalnym. Oznacza to, że przy takich samych parametrach ataku stawał się on mniej destruktywny. Dla wszystkich modeli CNN dokładność klasyfikacji w badanym punkcie pracy była wyższa, a więc skuteczności wszystkich ataków była niższa. Największy spadek skuteczności zaobserwowano dla modelu wyposażonego w warstwy generatywne szumu Cauchy'ego, najmniejszy zaś dla modelu wyposażonego w warstwy generatywne szumu białego o rozkładzie jednostajnym.

Ogólnie zaś należy zauważać, iż wykresy wzrostu energii perturbacji od parametru  $\epsilon$  zachowały swój ogólny kształt i pozostają bardzo zbliżone do bazowych wykresów, pochodzących z badań na modelu podstawowym. Jednakże zaobserwowano pewne wypłaszczenie wykresów spadku dokładności klasyfikacji w zależności od parametru  $\epsilon$ . Oznacza ono pogarszanie się skuteczności ataku, wraz ze wzrostem tegoż parametru. Wypłaszczenie to jest bardziej dostrzegalne dla modeli CNN, operujących na zbiorze danych Fashion MNIST.

### 4.3.2. Porównanie strategii regularyzacji szumem dla ataku *LESGSM*

Parametrem sterującym zakłócenia była wartość  $\epsilon$ , która wyznaczała bazę wzmocnienia kierunku gradientu funkcji straty. Badania przeprowadzono dla zakłóceń z zakresu  $\epsilon \in [0.00, 0.05]$  dla zbioru Classic MNIST, oraz  $\epsilon \in [0.00, 0.02]$  dla zbioru Fashion MNIST.



Rys. 4.8: Porównanie wyników poszczególnych prób badawczych dla ataku LESGSM

Energy at 80% accuracy			
dataset	avg. energy	$\Delta$ energy	regularization
Classic MNIST (DNN)	$\approx 4.6 \cdot 10^{-1}$	-29.2%	Uniform
Classic MNIST (DNN)	$\approx 5.5 \cdot 10^{-1}$	-15.4%	Normal
Classic MNIST (DNN)	$\approx 4.9 \cdot 10^{-1}$	-24.6%	Cauchy
Fashion MNIST (CNN)	$\approx 3.7 \cdot 10^{-2}$	+27.6%	Uniform
Fashion MNIST (CNN)	$\approx 3.0 \cdot 10^{-2}$	+3.45%	Normal
Fashion MNIST (CNN)	$\approx 5.0 \cdot 10^{-2}$	+72.4%	Cauchy

Tab. 4.9: Tabela zależności energii perturbacji oraz dokładności klasyfikacji dla serii badawczej

Accuracy at last $\epsilon$ parameter			
dataset	avg. accuracy	$\Delta$ accuracy	regularization
Classic MNIST (DNN)	$\approx 68\%$	$-0.73\%$	Uniform
Classic MNIST (DNN)	$\approx 71\%$	$+3.65\%$	Normal
Classic MNIST (DNN)	$\approx 69\%$	$+0.73\%$	Cauchy
Fashion MNIST (CNN)	$\approx 69\%$	$+10.4\%$	Uniform
Fashion MNIST (CNN)	$\approx 69\%$	$+10.4\%$	Normal
Fashion MNIST (CNN)	$\approx 71\%$	$+13.6\%$	Cauchy

Tab. 4.10: Tabela zależności dokładności klasyfikacji od ostatniego parametru  $\epsilon$  dla serii badawczej

## Wynik badania

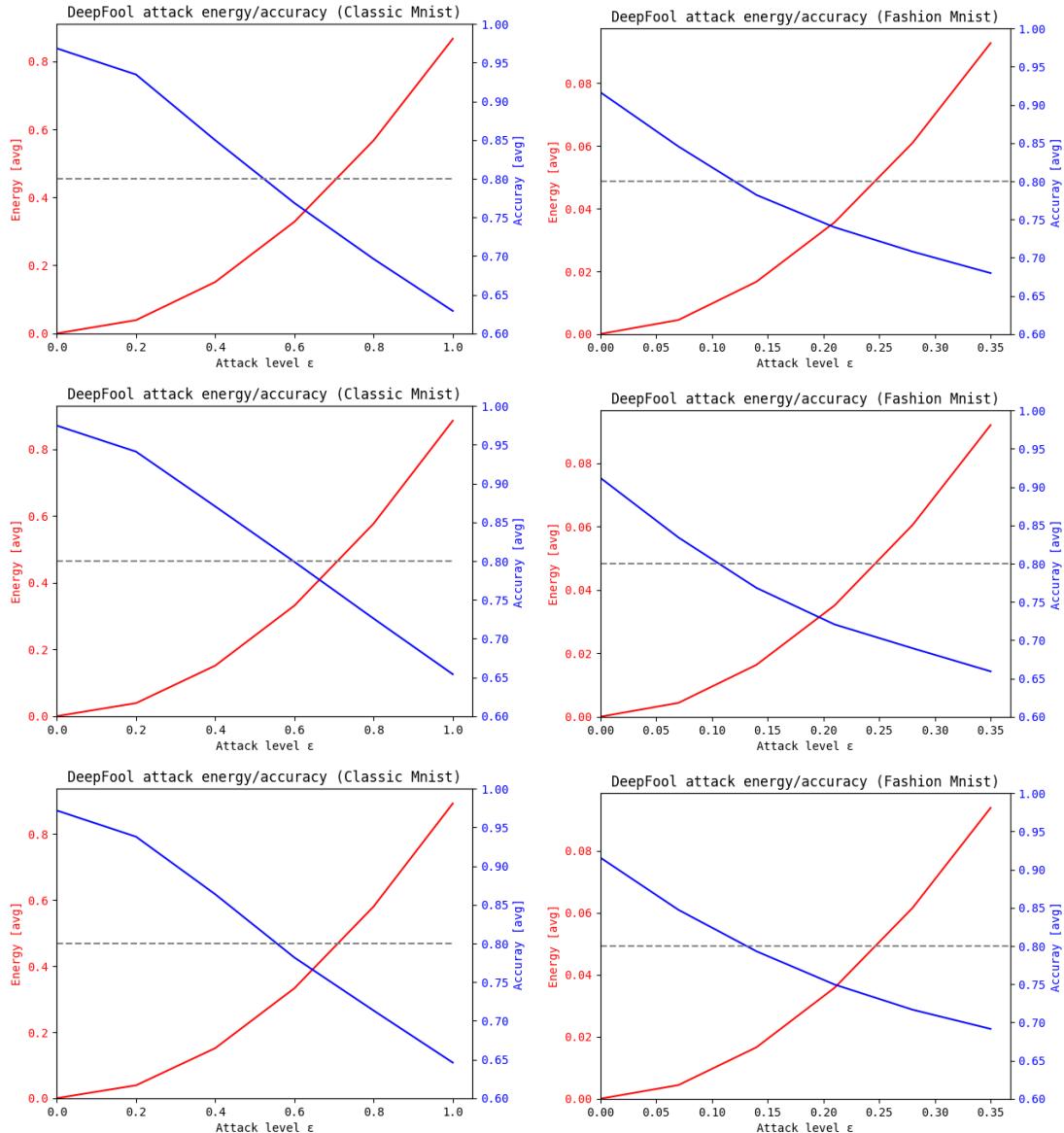
W punkcie pracy 80% dokładności klasyfikacji wykazano ogólną tendencję spadkową energii w dla modelu DNN operującego na zbiorze Classic MNIST, oraz ogólną tendencję wzrostową energii dla modelu CNN operującego na zbiorze Fashion MNIST. W przypadku modeli DNN wyposażonych w warstwy generatywne szumu białego o rozkładzie jednostajnym oraz szumu Cauchy'ego zmiany te były na podobnym poziomie, zaś dla modelu wyposażonego w warstwy generatywne szumu białego o rozkładzie normalnym były one zauważalnie mniejsze. W przypadku modeli CNN różnice te były istotnie większe. Najmniejszy wzrost energii klasyfikacji zanotowano dla modelu wyposażonego w warstwy generatywne szumu białego o rozkładzie normalnym, największe zaś dla modelu wyposażonego w warstwy generatywne szumu Cauchy'ego. Dla modelu wyposażonego w warstwy generatywne szumu białego o rozkładzie normalnym różnice te były na zbliżonym poziomie, do w modelu DNN. Patrząc zbiorczo, to odnotowane zmiany były doniosłejsze, w stosunku do analogicznych zmian dla algorytmu FGSM.

Końcowa dokładność klasyfikacji wszystkich modeli dla ostatniego badanego parametru  $\epsilon$  była na bardzo zbliżonym poziomie dla wszystkich modeli DNN dla modeli wyposażonych w warstwy generatywne szumu białego o rozkładzie jednostajnym oraz szumu Cauchy'ego. Zaobserwowano niewielki spadek skuteczności ataku dla modelu wyposażonego w warstwy generatywne szumu białego o rozkładzie normalnym. Oznacza to, że przy takich samych parametrach ataku stawał się on mniej destruktywny. Spadek ten był jednak mniejszy, aniżeli miało to miejsce w wypadku algorytmu FGSM. Dla wszystkich modeli CNN dokładność klasyfikacji w badanym punkcie pracy była wyższa, a więc skuteczności wszystkich ataków była niższa. Spadek ten był zbliżony poziomem do spadku skuteczności zaobserwowanego dla ataku FGSM. Nie zaobserwowano znaczących różnic ze względu na rodzaj warstw szumu generatywnego.

Ogólnie zaś należy zauważyć, iż wykresy wzrostu energii perturbacji od parametru  $\epsilon$  zachowały swój ogólny kształt i pozostają bardzo zbliżone do bazowych wykresów, pochodzących z badań na modelu podstawowym. Jednakże zaobserwowano pewne wypłaszczenie wykresów spadku dokładności klasyfikacji w zależności od parametru  $\epsilon$ . Oznacza ono pogarszanie się skuteczności ataku, wraz ze wzrostem tegoż parametru. Wypłaszczenie to jest bardziej dostrzegalne dla modeli CNN, operujących na zbiorze danych Fashion MNIST.

### 4.3.3. Porównanie strategii regularyzacji szumem dla ataku DeepFool

Parametrem sterującym zakłócenia była wartość  $\epsilon$ , która wyznaczała bazę wzmocnienia kierunku gradientu funkcji straty. Badania przeprowadzono dla zakłóceń z zakresu  $\epsilon \in [0.00, 1.00]$  dla zbioru Classic MNIST, oraz  $\epsilon \in [0.00, 0.35]$  dla zbioru Fashion MNIST.



Rys. 4.9: Porównanie wyników poszczególnych prób badawczych dla ataku DeepFool

Energy at 80% accuracy			
dataset	avg. energy	$\Delta$ energy	regularization
Classic MNIST (DNN)	$\approx 2.6 \cdot 10^{-1}$	-46.9%	Uniform
Classic MNIST (DNN)	$\approx 3.3 \cdot 10^{-1}$	-32.6%	Normal
Classic MNIST (DNN)	$\approx 3.0 \cdot 10^{-1}$	-38.7%	Cauchy
Fashion MNIST (CNN)	$\approx 1.3 \cdot 10^{-2}$	+30.0%	Uniform
Fashion MNIST (CNN)	$\approx 1.1 \cdot 10^{-2}$	+10.0%	Normal
Fashion MNIST (CNN)	$\approx 1.5 \cdot 10^{-2}$	+50.0%	Cauchy

Tab. 4.11: Tabela zależności energii perturbacji oraz dokładności klasyfikacji dla serii badawczej.

Accuracy at last $\epsilon$ parameter			
dataset	avg. accuracy	$\Delta$ accuracy	regularization
Classic MNIST (DNN)	$\approx 63\%$	$-7.35\%$	Uniform
Classic MNIST (DNN)	$\approx 65\%$	$-4.41\%$	Normal
Classic MNIST (DNN)	$\approx 65\%$	$-4.41\%$	Cauchy
Fashion MNIST (CNN)	$\approx 68\%$	$+9.68\%$	Uniform
Fashion MNIST (CNN)	$\approx 66\%$	$+6.45\%$	Normal
Fashion MNIST (CNN)	$\approx 69\%$	$+11.3\%$	Cauchy

Tab. 4.12: Tabela zależności dokładności klasyfikacji od ostatniego parametru  $\epsilon$  dla serii badawczej

## Wynik badania

W punkcie pracy 80% dokładności klasyfikacji wykazano ogólną tendencję spadkową energii w dla modelu DNN operującego na zbiorze Classic MNIST, oraz ogólną tendencję wzrostową energii dla modelu CNN operującego na zbiorze Fashion MNIST. W przypadku modeli DNN zmiany te były na podobnym poziomie, jednakże zauważalnie największe dla modelu wyposażonego w warstwy generatywne szumu białego o rozkładzie jednostajnym, a najmniejsze dla modelu wyposażonego w warstwy generatywne szumu białego o rozkładzie normalnym. W przypadku modeli CNN różnice te były istotnie większe. Najmniejszy wzrost energii klasyfikacji zanotowano dla modelu wyposażonego w warstwy generatywne szumu białego o rozkładzie normalnym, największe zaś dla dla modelu wyposażonego w warstwy generatywne szumu Cauchy'ego. Patrząc zbiorczo, to odnotowane zmiany były donioślejsze, w stosunku do analogicznych zmian dla algorytmu FGSM; większe nawet niż w przypadku algorytmu LESGSM.

Końcowa dokładność klasyfikacji modeli dla ostatniego badanego parametru  $\epsilon$  była zauważalnie niższa dla wszystkich badanych modeli DNN. Największy wzrost skuteczności ataku zaobserwowanego dla modelu wyposażonego w warstwy generatywne szumu białego o rozkładzie jednostajnym. W przypadku pozostałych modeli był on istotnie mniejszy i na podobnym poziomie. Oznacza to, że przy takich samych parametrach ataku stawał się on bardziej destruktywny. Dla wszystkich modeli CNN dokładność klasyfikacji w badanym punkcie pracy była wyższa, a więc skuteczności wszystkich ataków była niższa. Spadek ten był zbliżony poziomem do spadku skuteczności zaobserwowanego dla ataku FGSM, mniejszy jednak aniżeli w wypadku ataku LESGSM.

Ogólnie zaś należy zauważyć, iż wykresy wzrostu energii perturbacji od parametru  $\epsilon$  zachowały swój ogólny kształt i pozostają bardzo zbliżone do bazowych wykresów, pochodzących z badań na modelu podstawowym. Podobnie jak w dla ataków FGSM i LESGSM zaobserwowano pewne wypłaszczenie wykresów spadku dokładności klasyfikacji w zależności od parametru  $\epsilon$  dla modeli CNN. Równolegle jednak stwierdzono większą liniowość i szybszy spadek krzywej dokładności klasyfikacji dla modeli DNN, operujących na zbiorze Classic MNIST.

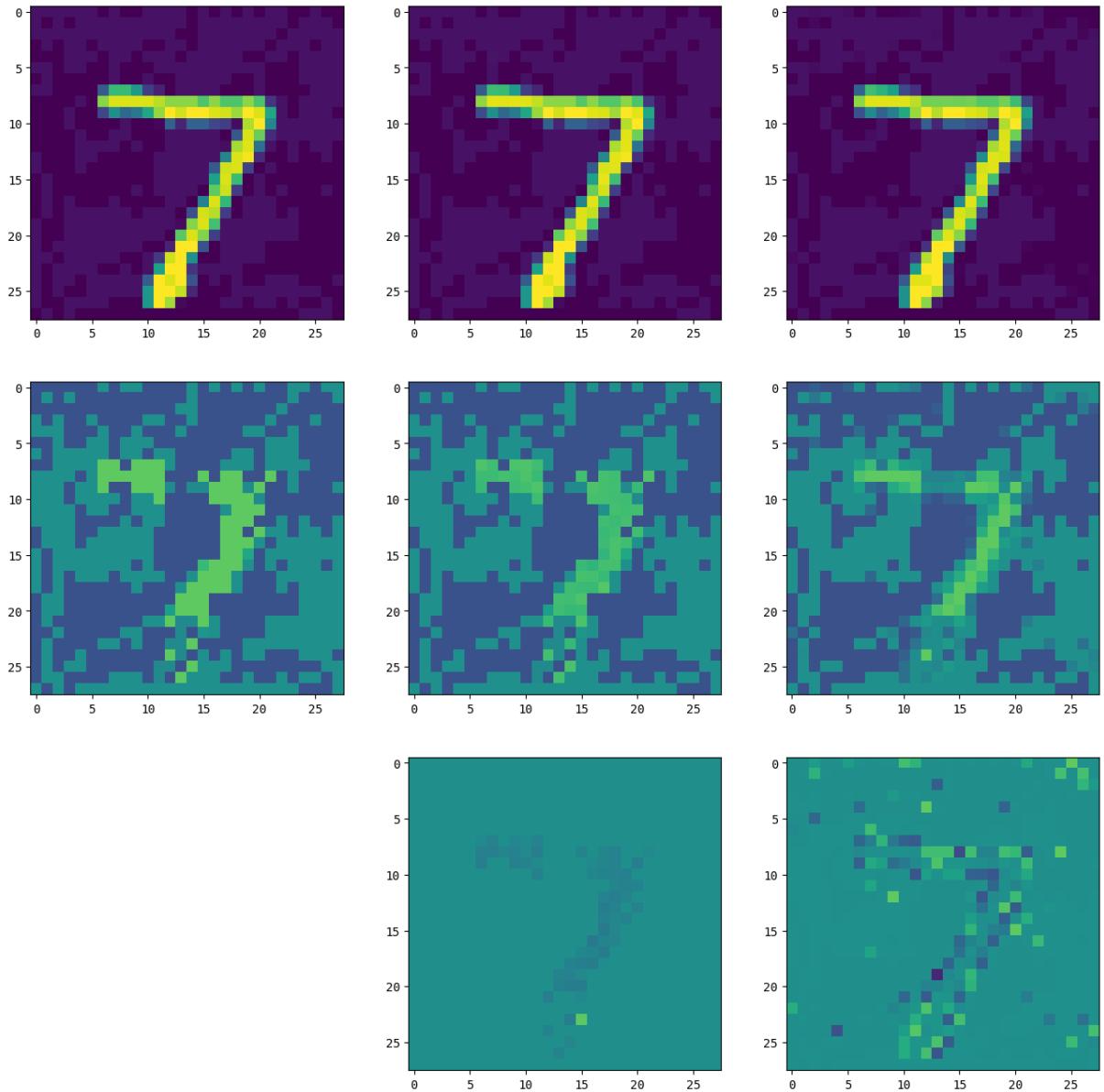
#### **4.3.4. Konkluzja wyników analizy symulacyjnej**

Badania wykazały, że modele DNN i CNN wykazują odmienną charakterystykę pracy, przy zastosowaniu warstw regularyzacji szumem generatywnym. Modele DNN operujące na zbiorze danych Classic MNIST wykazywały ogólnie zwiększenie podatności na zakłócenia kontradyktoryjne, względem modeli podstawowych. Niemalże w każdym badaniu wykazano spadek energii perturbacji wymaganej dla osiągnięcia dokładności klasyfikacji na poziomie 80%. Najmniejsze spadki zanotowano dla techniki regularyzacji szumem białym o rozkładzie normalnym. W wypadku ataku FGSM osiągał on bardzo zbliżone wyniki, co tradycyjna technika regularyzacji warstwami spatkowymi. Dla ataków FGSM i LESGSM nie zanotowano istotnego wzrostu dokładności klasyfikacji (a więc pogorszenia się skuteczności ataków), jednakże dla odnotowano zwiększenie skuteczności ataku DeepFool. Dla tego samego parametru  $\epsilon$  spadek dokładności klasyfikacji był większy, aniżeli w modelu podstawowym.

W wypadku modeli CNN operujących na zbiorze danych Fashion MNIST stwierdzono ogólnie zmniejszenie podatności na zakłócenia kontradyktoryjne, względem modeli podstawowych. Wzrost energii perturbacji konieczny do osiągnięcia dokładności klasyfikacji na poziomie 80% wzrósł dla ataków LESGSM oraz DeepFool, zwłaszcza dla modeli wyposażonych w warstwy generatywne szumu białego o rozkładzie jednostajnym oraz szumu Cauchy'ego. W wypadku ataku FGSM wzrost ten był istotnie mniejszy oraz miał zastosowanie tylko w wypadku szumu Cauchy'ego. Dla każdego z ataków stwierdzono wzrost dokładności klasyfikacji, przy zachowaniu tego samego parametru  $\epsilon$  co w modelu podstawowym. Wzrost ten był na podobnym poziomie we wszystkich badanych modelach, co oznacza, iż wszystkie algorytmy wykazywały mniejszą destruktywność względem modelu podstawowego.

### **4.4. Ocena efektywności algorytmu LESGSM**

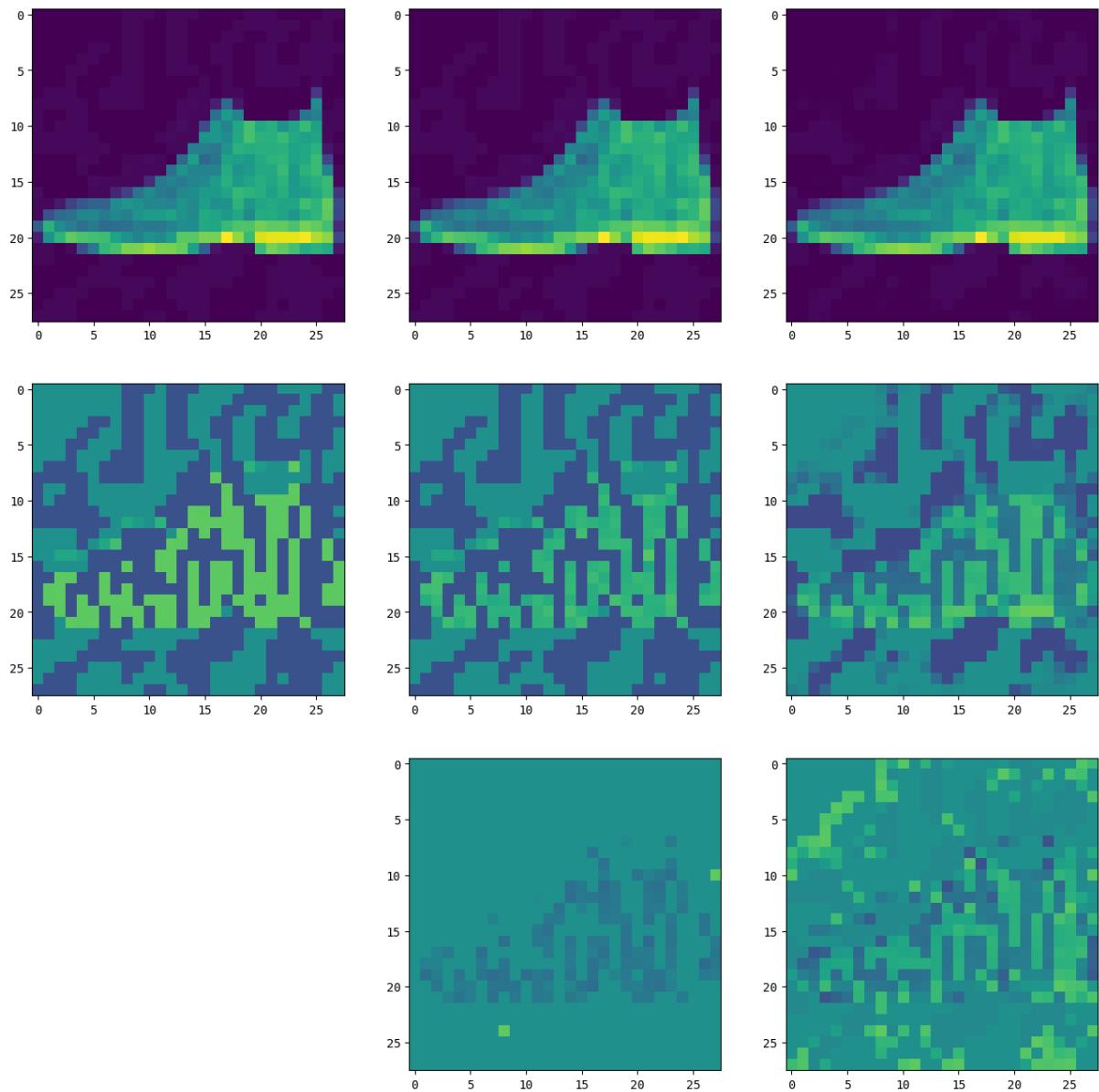
Przeprowadzono badania porównawcze algorytmów FGSM, LESGSM oraz DeepFool. Dla algorytmów FGSM oraz DeepFool parametry dobrano tak, aby zapewnić około 62% dokładności klasyfikacji ( $\pm 2$  p.p.). W przypadku ataku LESGSM parametr  $\epsilon$  był identyczny jak w algorytmie FGSM, co miało na celu zapewnienie najwyższej precyzji porównania obu algorytmów. W badaniach mierzono energię perturbacji oraz wynikową dokładność klasyfikacji, a następnie przeprowadzono zestawienie i porównanie procentowych różnic między poszczególnymi atakami. Wygenerowano również poglądowe obrazy z zakłóceniami, obrazy samych zakłóceń, a także obraz różnic w zakłóciach za pomocą algorytmów LESGSM i DeepFool względem podstawowej metody FGSM. Podglądy te są pomocne w kontekście wizualnej oceny różnic w perturbacjach.



Rys. 4.10: Przykładowe powidoki zakłóceń kontradiktoryjnych na zbiorze *Classic MNIST*. W siatce od lewej: atak FGSM, atak LESGSM, atak DeepFool. W siatce od góry: podgląd obrazu z zakłóceniem, podgląd wyłącznie zakłócenia obrazu, podgląd różnic w zakłóceniu w stosunku do zakłócenia FGSM [opracowanie własne]

Classic MNIST (DNN) at 62% accuracy ( $\pm 2$ p.p.)		
attack	$\Delta$ energy	accuracy
FGSM	+5.97%	61.51%
LESGSM	...	63.46%
DeepFool	-13.44%	62.54%

Tab. 4.13: Tabela porównawcza efektywności ataków na zbiorze *Classic MNIST*



Rys. 4.11: Przykładowe powidoki zakłóceń kontradydaktryjnych na zbiorze *Fashion MNIST*. W siatce od lewej: atak FGSM, atak LESGSM, atak DeepFool. W siatce od góry: podgląd obrazu z zakłóceniem, podgląd wyłącznie zakłócenia obrazu, podgląd różnic w zakłóceniu w stosunku do zakłócenia FGSM [opracowanie własne]

Fashion MNIST (CNN) at 62% accuracy ( $\pm 2$ p.p.)		
attack	$\Delta$ energy	accuracy
FGSM	+25.63%	62.03%
LESGSM	...	61.76%
DeepFool	-48.95%	59.24%

Tab. 4.14: Tabela porównawcza efektywności ataków na zbiorze *Fashion MNIST*

## **Wynik badania**

Algorytm LESGSM okazał się skuteczny w realizacji swoich założeń. Analizując energię perturbacji uzyskaną dla algorytmu LESGSM w odniesieniu do innych metod, można zauważyc, że dla modelu DNN operującego na zbiorze danych Classic MNIST, energia perturbacji była o około 6% wyższa dla algorytmu FGSM i około 13.5% niższa dla algorytmu DeepFool, przy zachowaniu zakładanej tolerancji ( $\pm 2 \text{ p.p.}$ ) względem algorytmu FGSM. Dla modelu CNN operującego na zbiorze danych Fashion MNIST różnice te były jeszcze bardziej znaczące: energia perturbacji była o około 25.5% wyższa dla algorytmu FGSM i około 49% niższa dla algorytmu DeepFool. Co więcej, algorytm LESGSM wykazał wyższą skuteczność ataku w porównaniu z algorytmem FGSM, na którym jest oparty.

Dodatkowo przeprowadzono badania dotyczące powidoków zakłóceń, które polegały na analizie różnic pomiędzy perturbacjami generowanymi przez poszczególne algorytmy. Wyniki tych badań okazały się zaskakujące; różnica perturbacji  $\delta_{\text{FGSM}} - \delta_{\text{LESGSM}}$  była wizualnie niezwykle podobna do różnicy  $\delta_{\text{FGSM}} - \delta_{\text{DeepFool}}$ . Zjawisko to zasługuje na szczególną uwagę, ponieważ sugeruje istnienie pewnych wspólnych cech pomiędzy perturbacjami generowanymi przez metody LESGSM i DeepFool, mimo że oba algorytmy różnią się pod względem założeń i implementacji. Na podstawie analizy wizualnej zauważono, że powidoki perturbacji generowanych przez FGSM oraz LESGSM wykazują niezwykle podobny układ pikseli, których wartość została obniżona.

# Rozdział 5

## Podsumowanie

Badania przeprowadzone w niniejszej pracy zorientowane były na zrozumienie wpływu zakłóceń stochastycznych i kontradyktoryjnych na skuteczność modeli neuronowych w klasyfikacji obrazów. Przeanalizowano trzy typy zakłóceń stochastycznych: szum biały o rozkładzie jednostajnym, szum biały o rozkładzie normalnym, szum Cauchy'ego, i trzy typy zakłóceń kontradyktoryjnych: atak FGSM, atak DeepFool oraz autorki atak LESGSM. Dodatkowo przebadano także strategię regularyzacji szumem w kontrze do tradycyjnej techniki regularyzacji spadkowej (ang. *dropout*). Zaproponowano trzy modele z zastosowaniem warstw generatywnych szumu, które generowały 100% zadanego poziomu szumu w trakcie treningu oraz 25% w trakcie normalnej aktywności. Przebadano trzy rodzaje warstw: szumu białego o rozkładzie jednostajnym, szumu białego o rozkładzie normalnym oraz szumu Cauchy'ego.

Pierwsza część badań dotyczyła wpływu zakłóceń stochastycznych. Wyniki wykazały, że każdy rodzaj szumu ma odmienny wpływ na modele neuronowe. Szum biały o rozkładzie jednostajnym okazał się najbardziej destruktywny, a więc wymagał najmniejszej energii perturbacji do obniżenia dokładności klasyfikacji do zadanego poziomu. Za nim plasował się szum biały o rozkładzie normalnym, który wymagał nieco wyższych poziomów energii ( $\approx 10\%$ ). Szum Cauchy'ego zdawał się najbardziej zniekształcać obraz, lecz wymagał ponad dwukrotnie wyższych poziomów energii perturbacji, aby osiągnąć pożądany spadek dokładności klasyfikacji. Ponadto spostrzeżono, iż model DNN operujący na zbiorze danych Classic MNIST był około  $10 \times$  mniej podatny na wpływ zakłóceń, aniżeli model CNN operujący na zbiorze danych Fashion MNIST.

Druga część badań skupiła się na zakłócenach kontradyktoryjnych, generowanych przez ataki FGSM (*Fast Gradient Sign Method*), LESGSM (*Low Energy Stochastic Gradient Sign Method*) oraz DeepFool. Badania te miały na celu ocenę skuteczności każdego z tych ataków w obniżaniu dokładności modeli neuronowych. Najbardziej destruktywnym okazał się atak DeepFool, który najskuteczniej obniżał dokładność klasyfikacji obrazów. Ataki FGSM i LESGSM również wykazyły wysoką skuteczność, choć nieco niższą w porównaniu do DeepFool. Ponadto spostrzeżono, iż model DNN operujący na zbiorze danych Classic MNIST był około  $10 \times$  mniej podatny na wpływ zakłóceń, aniżeli model CNN operujący na zbiorze danych Fashion MNIST.

Trzecia część badań poświęcona była analizie skuteczności różnych strategii regularyzacji szumem w obniżaniu skuteczności ataków kontradyktoryjnych. W wypadku modeli DNN operujących na zbiorze danych Classic MNIST żadna z zaproponowanych strategii nie okazała się skuteczna. Co więcej, w wypadku ataku DeepFool istotnie pogarszały one odporność modeli. Zanotowano także mniejszy spadek odporności w wypadku ataków FGSM i LESGSM. Dla modeli CNN operujących na zbiorze danych Fashion MNIST strategie te okazały się skuteczne. Największą skuteczność wykazały one wobec ataków DeepFool oraz LESGSM. W przypadku ataku FGSM skuteczność ta była o wiele mniejsza, wręcz marginalna. Największy wzrost odporności modeli zanotowano dla strategii regularyzacji szumem Cauchy'ego, a w drugiej kolejności szumem białym o rozkładzie jednostajnym. W wypadku strategii regularyzacji szumem białym o rozkładzie normalnym wzrost odporności modeli występował, lecz był on marginalny.

Na koniec przeprowadzono szczegółowe badania porównawcze autorskiego algorytmu kontradyktoryjnego LESGSM z atakami FGSM i DeepFool. Wykazano skuteczność zastosowanego podejścia, zarówno w obniżaniu energii perturbacji, jak i zachowaniu podobnego poziomu skuteczności jak algorytm FGSM. Algorytm wykazuje większą skuteczność w realizacji swoich zadań dla modeli CNN operujących na zbiorze Fashion MNIST. Przeprowadzono także analizy powidoków zakłóceń, które pokazały, że różnica perturbacji  $\delta_{\text{FGSM}} - \delta_{\text{LESGSM}}$  była wizualnie podobna do różnicy  $\delta_{\text{FGSM}} - \delta_{\text{DeepFool}}$ . To sugeruje istnienie wspólnych cech perturbacji generowanych przez LESGSM i DeepFool, mimo różnic w ich założeniach i implementacji.

Niniejsza praca może stanowić punkt wyjścia do dalszych badań na wielu płaszczyznach. Przede wszystkim, istotne jest zbadanie wpływu zakłóceń stochastycznych i kontradyktoryjnych w zastosowaniach innych niż klasyfikacja obrazów, takich jak analiza sentymetu, rozpoznanie mowy oraz przetwarzanie języka naturalnego. Na podstawie algorytmu LESGSM warto również rozważyć opracowanie nowych algorytmów kontradyktoryjnych, które charakteryzują się obniżoną energią perturbacji, ale jednocześnie są zdolne do przeprowadzania ataków w scenariuszach czarnej skrzynki. Ponadto należałoby przeprowadzić ewaluację techniki LESGSM w kontekście innych rodzajów klasyfikatorów, takich jak modele probabilistyczne czy maszyny wektorów nośnych. Innym wartym rozważenia kierunkiem badawczym jest analiza zastosowania przedstawionych technik regularyzacji szumem do innych architektur modeli neuronowych, takich jak sieci rekurencyjne (RNN) czy transformatory. Dodatkowo, badania długoterminowego wpływu przedstawionych technik regularyzacji szumem na stabilność i generalizację modeli neuronowych, a także rozszerzenie badań na większe i bardziej zróżnicowane zbiorы danych.

# Literatura

- [1] I. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in *International Conference on Learning Representations*, p. 20, ICLR, 2015.
- [2] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, “Deepfool: a simple and accurate method to fool deep neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2574–2582, IEEE, 2016.
- [3] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [4] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *The bulletin of mathematical biophysics*, vol. 5, pp. 115–133, 1943.
- [5] P. Kuznetsov, R. Edmunds, T. Xiao, H. Iqbal, R. Puri, N. Golmant, and S. Shih, “Kontradyktryjne uczenie maszynowe,” *Napędy i Sterowanie*, vol. 23, no. 7/8, pp. 80–89, 2021.
- [6] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, “The limitations of deep learning in adversarial settings,” in *2016 IEEE European symposium on security and privacy (EuroS&P)*, pp. 372–387, IEEE, 2016.
- [7] T. Forrest, “From sender to receiver: propagation and environmental effects on acoustic signals,” *American zoologist*, vol. 34, no. 6, pp. 644–654, 1994.
- [8] S. Caldara, S. Nuccio, and C. Spataro, “Measurement uncertainty estimation of a virtual instrument,” in *Proceedings of the 17th IEEE Instrumentation and Measurement Technology Conference [Cat. No. 00CH37066]*, vol. 3, pp. 1506–1511, IEEE, 2000.
- [9] I. G. Vladimirov and P. Diamond, “A uniform white-noise model for fixed-point roundoff errors in digital systems,” *Automation and Remote Control*, vol. 63, pp. 753–765, 2002.
- [10] W. Liu and W. Lin, “Additive white gaussian noise level estimation in svd domain for images,” *IEEE Transactions on Image processing*, vol. 22, no. 3, pp. 872–883, 2012.
- [11] F. Sciacchitano, Y. Dong, and T. Zeng, “Variational approach for restoring blurred images with cauchy noise,” *SIAM Journal on Imaging Sciences*, vol. 8, no. 3, pp. 1894–1922, 2015.

- [12] C. R. Steffens, L. R. V. Messias, P. L. J. Drews, and S. S. da Costa Botelho, “Can exposure, noise and compression affect image recognition? an assessment of the impacts on state-of-the-art convnets,” in *2019 Latin American Robotics Symposium (LARS), 2019 Brazilian Symposium on Robotics (SBR) and 2019 Workshop on Robotics in Education (WRE)*, pp. 61–66, IEEE, 2019.
- [13] S. Dodge and L. Karam, “Understanding how image quality affects deep neural networks,” in *2016 eighth international conference on quality of multimedia experience (QoMEX)*, pp. 1–6, IEEE, 2016.
- [14] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, “A survey on adversarial attacks and defences,” *CAAI Transactions on Intelligence Technology*, vol. 6, no. 1, pp. 25–45, 2021.
- [15] J. Lin, L. Xu, Y. Liu, and X. Zhang, “Black-box adversarial sample generation based on differential evolution,” *Journal of Systems and Software*, vol. 170, p. 110767, 2020.
- [16] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, “Practical black-box attacks against machine learning,” in *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pp. 506–519, 2017.
- [17] J. Sen and S. Dasgupta, “Adversarial attacks on image classification models: Fgsm and patch attacks and their impact,” in *Information Security and Privacy in the Digital World—Some Selected Topics*, IntechOpen, 2023.
- [18] A. Morgan, “A review of deepfool: a simple and accurate method to fool deep neural networks — medium.com.” <https://medium.com/machine-intelligence-and-deep-learning-lab/a-review-of-deepfool-a-simple-and-accurate-method-to-fool-deep-neural-networks-b016fba9e48e>, 2022. [Accessed 01-06-2024].
- [19] Y. LeCun, C. Cortes, and C. Burges, *MNIST handwritten digit database*. R Foundation for Statistical Computing, 1994. [Accessed 01-06-2024].
- [20] H. Xiao, K. Rasul, and R. Vollgraf, *Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms*. Zalando Research, 2017. [Accessed 01-06-2024].
- [21] “Training a neural network on mnist with keras.” [https://www.tensorflow.org/datasets/keras\\_example](https://www.tensorflow.org/datasets/keras_example), 2023. [Accessed 01-06-2024].
- [22] “Convolutional neural network (cnn).” <https://www.tensorflow.org/tutorials/images/cnn>, 2024. [Accessed 01-06-2024].