

Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI  
I TECHNIK INFORMACYJNYCH



Instytut Automatyki i Informatyki Stosowanej

# Praca dyplomowa magisterska

na kierunku Informatyka  
w specjalności Inteligentne systemy

Badanie możliwości tworzenia wzorów użytkowych  
za pomocą sieci neuronowych na przykładzie szydełkowania

**Małgorzata Wróbel**

Numer albumu 289337

promotor  
dr hab. inż. Mariusz Kamola

WARSZAWA 2023

*Pragnę wyrazić wdzięczność  
Panu Doktorowi hab. inż. Mariuszowi Kamoli  
za opiekę i wsparcie podczas pisania niniejszej pracy.*

## **Badanie możliwości tworzenia wzorów użytkowych za pomocą sieci neuronowych na przykładzie szydełkowania**

**Streszczenie.** W ostatnich latach sztuczna inteligencja znajduje zastosowanie w coraz większej liczbie dziedzin życia człowieka i jest już wykorzystywana nawet w sztuce. Mimo to sposoby użycia jej metod do wspierania rękodzielnictwa wciąż są niezbadane. Dostępna nieliczna literatura porusza możliwości połączenia uczenia maszynowego i tworzenia wyrobów na drutach. Szydełkowanie natomiast jest niemożliwe do podrobienia przez maszynę, zatem sztuczna inteligencja mogłaby posłużyć do wspierania procesu twórczego dziewiarzy.

Celem niniejszej pracy było zaproponowanie i zbadanie metody tworzenia graficznych schematów szydełkowych w formie kwadratu (ang. *granny squares*) za pomocą autoenkoderów wariacyjnych. Sieci te stanowią klasę modeli statystycznych ze zmiennymi ukrytymi. Zastosowanie kodowania obrazów we współrzędnych kartezjańskich i biegunowych umożliwia uzyskanie różnych reprezentacji cech wzorów szydełkowych. Eksperymenty z przygotowaną siecią neuronową obejmują wykorzystanie dwóch rodzajów dekoderek do generowania nowych obrazów: dekodera Bernoulliego (obrazy zero-jedynkowe) i dekodera Gaussa (obrazy ze skalą szarości). Istotną część tejże pracy stanowi stworzony specjalnie na jej potrzeby zbiór danych składający się z 2,5 tys. obrazów przekształconych do postaci binarnej, przedstawiających schematy kwadratów szydełkowych.

Wykorzystanie metod sztucznej inteligencji pozwala na generowanie nowych schematów szydełkowych z różnym stopniem rozmycia szczegółów. Zaproponowane kodowanie danych we współrzędnych biegunowych, a w szczególności pobieranie wielu wycinków informacji w postaci kafelków z obrazów w tym układzie, wzbogaca zbiór danych i skutkuje generacją urozmaiconych wzorów.

**Słowa kluczowe:** autoenkodery wariacyjne, szydełkowanie, układ współrzędnych biegunowych, współrzędne biegunowe, wzbogacanie danych

## **Investigation of the possibility of creating utility patterns using neural networks by the example of crochet**

**Abstract.** In recent years artificial intelligence has found its way into more and more areas of human life and is already being used even in arts. Nevertheless, the ways in which its methods could be used to support handicraft have not yet been investigated. The sparse literature available touches upon the ways of merging machine learning and knitting. Crochet, however, is impossible to counterfeit by machine, therefore artificial intelligence could serve to aid the creative process of crocheters.

The aim of this thesis was to propose and explore the methods of creating graphic granny square crochet patterns with variational autoencoders. These networks belong to a family of latent variable models. The usage of image encoding in Cartesian and polar coordinate systems helps obtain varied feature representations of crochet patterns. Experiments with the prepared neural network include the use of two types of decoders for new image generation: Bernoulli decoder (binary images) and Gaussian decoder (grayscale images). An important part of this work is a dataset created particularly for the purpose, consisting of 2,500 binarized images depicting crochet granny square patterns.

The use of artificial intelligence provides the opportunity for generation of new crochet patterns with varying level of blurriness. The proposed encoding of data in polar coordinates, and in particular the extraction of multiple tile-like segments of information from images in this system, augments the dataset and results in the generation of diversified patterns.

**Keywords:** variational autoencoders, crochet, polar coordinates, polar coordinate system, data augmentation

# Spis treści

<b>1. Wstęp</b> . . . . .	6
<b>2. Cel i zakres pracy</b> . . . . .	7
<b>3. Technika szydełkowania i dostępne dane</b> . . . . .	8
3.1. Podstawy techniki szydełkowania . . . . .	8
3.2. Dostępne dane . . . . .	13
<b>4. Zaproponowane rozwiązanie</b> . . . . .	17
4.1. Przegląd literatury . . . . .	17
4.2. Metodyka . . . . .	17
4.3. Kodowanie danych . . . . .	18
4.4. Sieć typu autoenkoder wariacyjny . . . . .	20
<b>5. Eksperymenty</b> . . . . .	24
5.1. Model sieci neuronowej i sposób uczenia . . . . .	24
5.2. Augmentacja danych . . . . .	25
5.3. Eksperyment z grafikami w układzie współrzędnych kartezjańskich . . . . .	26
5.4. Eksperyment z grafikami w układzie współrzędnych biegunowych . . . . .	26
5.5. Eksperyment z pominięciem niekorzystnych zjawisk w układzie współrzędnych biegunowych . . . . .	27
5.6. Eksperyment z kawałkami grafik w układzie współrzędnych biegunowych . . . . .	27
5.7. Eksperymenty z kawałkami grafik w układzie współrzędnych kartezjańskich . . . . .	29
<b>6. Wyniki</b> . . . . .	30
<b>7. Wnioski</b> . . . . .	39
<b>Bibliografia</b> . . . . .	41
<b>Wykaz symboli i skrótów</b> . . . . .	43
Symbole matematyczne . . . . .	43
Skróty . . . . .	43
<b>Spis rysunków</b> . . . . .	43
<b>Spis tabel</b> . . . . .	46
<b>Spis dodatków</b> . . . . .	46

# 1. Wstęp

Metody sztucznej inteligencji znajdują zastosowanie w kolejnych dziedzinach życia. Mimo licznych kontrowersji coraz częściej są wykorzystywane do tworzenia dzieł artystycznych. Dzisiaj profesjonalne programy graficzne oferują narzędzia wspomagania projektowego wykorzystującego modele generatywne, których działanie polega np. na uzupełnianiu zaznaczonego pola automatycznie sfabrykowaną treścią. Zapotrzebowanie na treść (ang. *content*) stale rośnie, a granica pomiędzy określeniem „artysta” a „twórca contentu” zaciera się. Nie stanowi zaskoczenia zatem, że połączenie mechaniki i budowy maszyn ze sztuczną inteligencją jest już w stanie wyręczyć człowieka nawet w tworzeniu sztuki użytkowej i rękodzielnictwie.

Przykładem zmechanizowanego rzemiosła jest dziewiarstwo. Pierwsza maszyna przeznaczona do wytwarzania dzianiny została wynaleziona już w XVI wieku. Ręczne robienie na drutach jest łatwe do podrobienia przez maszynę, ponieważ polega na tworzeniu prostych pętelek i przekładaniu ich z jednego drutu na drugi, aż do „przerobienia” całego rzędu. Maszyna wykonuje podobną operację, ale na wielu haczykach jednocześnie, dzięki czemu jest w stanie wykonać cały nowy rząd pętelek w trakcie jednego kroku. Bez względu na to, czy proces jest ręczny czy maszynowy, kolejne oczka powstają w dwóch wymiarach i są zaczepiane w prosty sposób w pętelki z poprzedniego rzędu.

Istnieją jednak takie rodzaje rękodzielnictwa, które nie zostały zmechanizowane, jak szydełkowanie. Jest to proces wyrobu dzianiny za pomocą haczykowego narzędzia – szydełka – polegający na przeciąganiu włóczki przez pętelki. Każda kolejna wykonana pętelka jest „zamknięta” – zdejmuje się ją z szydełka (na drutach nowe pętelki są cały czas nałożone na druty). Do jej utworzenia jednak konieczny jest szereg ruchów w trzech wymiarach, a w poprzedni rząd można ją zaczepić na różne sposoby. Do tej pory nie powstały maszyny, które byłyby w stanie tworzyć robótki szydełkowe.

Szydełkuje się na podstawie instrukcji słownych, ale wciąż najlepszym i najchętniej wybieranym przedstawieniem kolejności wykonywania oczek są schematy obrazkowe. Niewątpliwym ich atutem jest to, że dają konkretne wyobrażenie o całym, gotowym wyrobie, inaczej niż instrukcje słowne, które przedstawiają kolejne polecenia.

Łatwość mechanizowania techniki robienia na drutach może wyjaśniać zainteresowanie tym procesem w środowisku naukowym. W ostatnim czasie oprócz amatorskich projektów pojawiają się także badania pod nadzorem uczelni akademickich, które służą przygotowaniu modeli sztucznej inteligencji do generowania instrukcji do robienia na drutach.

Pomimo rosnącej popularności szydełkowania brakuje projektów, literatury i artykułów naukowych o wykorzystaniu sztucznych sieci neuronowych i uczenia maszynowego do wspierania procesu wytwarzania dzianiny szydełkowej.

## 2. Cel i zakres pracy

Celem niniejszej pracy było badanie możliwości tworzenia wzorów użytkowych za pomocą sieci neuronowych na przykładzie szydełkowania na podstawie wzorów obrazkowych. Pomimo rosnącego zainteresowania tym rodzajem dziewiarstwa, wciąż brakuje publikacji o wykorzystaniu modeli sztucznej inteligencji do wspierania projektowania i wytwarzania wyrobów szydełkowych.

Badanie ograniczono do przetwarzania schematów kwadratów szydełkowych (z ang. *granny squares*) w związku z ich popularnością i dużymi zaletami w porównaniu do instrukcji słownych. Złożoność zagadnienia generacji takich wzorów, w porównaniu do wzorów na czapki i swetry, jest dużo mniejsza, co pozwoliło na otrzymanie zadowalających wyników za pomocą zaproponowanego modelu sieci neuronowej.

Z uwagi na brak gotowego zbioru danych uczących dla uczenia maszynowego ważną część pracy stanowiło skompletowanie i wstępne przetworzenie odpowiednich obrazów znalezionych w Internecie.

Wykorzystana w badaniach sztuczna sieć neuronowa to autoenkoder wariacyjny z warstwami splotowymi należący do klasy modeli statystycznych ze zmiennymi ukrytymi. Aktualny stan techniki oferuje całą gamę wnikliwie udokumentowanych modeli sieci neuronowych ogólnego i specjalnego przeznaczenia. Z użyciem otwartoźródłowych bibliotek programistycznych można uzyskać zadowalające wyniki nawet przy zastosowaniu stosunkowo prostych koncepcyjnie algorytmów uczenia maszynowego. W świetle tego niniejsza praca kładzie nacisk przede wszystkim na obróbkę wstępną danych wejściowych i przetwarzanie wyników, a nie na szczegółowe dostrajanie parametrów zaimplementowanego modelu.

Tworzenie kwadratów szydełkowych polega na dodawaniu kolistych rzędów wokół środka (początku) robótki. Powstałe w ten sposób obiekty przyjmują różne kształty w zależności od wybranego układu współrzędnych. W kolejnych rozdziałach poruszono zalety uczenia sieci neuronowej za pomocą obrazów przekształconych do współrzędnych biegunowych, a także różnych ich wycinków. Przedstawiono porównanie uzyskanych wyników z rezultatami uczenia z użyciem grafik we współrzędnych kartezjańskich.

Charakter wybranego modelu sieci neuronowej decyduje o jakości, ostrości i czytelności generowanych grafik – z tej przyczyny nie stanowią one kompletnych schematów kwadratów szydełkowych. Nie umniejsza to oczywiście ich wartości i przydatności. Zdecydowanie mogą one posłużyć jako źródło inspiracji do tworzenia nowych wzorów. We wnioskach przedstawiono kilka pomysłów, które mogłyby stanowić rozszerzenie czy kontynuację niniejszej pracy w celu uzyskania jeszcze wyższej jakości wyników.

W niniejszej pracy termin schemat i wzór używane są zamiennie do określenia instrukcji graficznej do wykonania robótki szydełkowej.

### 3. Technika szydełkowania i dostępne dane

#### 3.1. Podstawy techniki szydełkowania

Historycy debatują nad prawdziwym pochodzeniem szydełkowania. Technika znana w obecnej formie miała swój początek w XV-XVII wieku. Niewykluczone, że wcześniej prymitywna jej forma służyła na przykład do naprawiania sieci rybackich. Za pomocą prostego, haczykowatego narzędzia umożliwiającego ruchy w dowolnym kierunku oraz włączki można tworzyć dowolny rodzaj wyrobu z dzianiny. Z jednej, ciągłej nici (bez przerywania) dziewiarz jest w stanie wykonać kompletne przedmioty użytkowe w postaci czapek, szalików, serwet, a nawet swetrów. Alternatywna metoda tworzenia złożonych projektów szydełkowych polega na łączeniu ze sobą wielu mniejszych (przeważnie) kwadratowych elementów (ang. *granny squares*). W rezultacie otrzymana całość składa się ze wzorzystych, ażurowych lub pełnych kafelków.



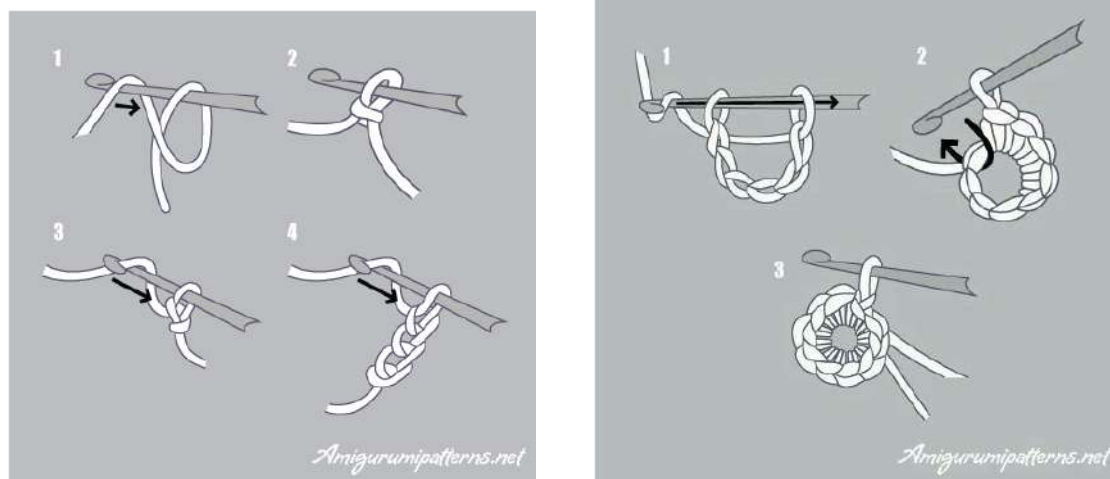
**Rysunek 3.1.** Sposób wykonywania kolorowego kwadratu szydełkowego z kwiatowym wizerunkiem (*Boho granny square pattern*, 2019)

Rysunek 3.1 pokazuje, że kwadraty szydełkowe powstają od środka, okrężnie w kierunku przeciwnym do ruchu wskazówek zegara. Sposób ten prowadzi do wytworzenia charakterystycznych, kolistych rzędów wokół środka. Każdy z nich jest kończony specjalnym oczkiem zamykającym (tzw. oczkiem ścisłym, ang. *slip stitch*). Następne okążenie zaczyna się od wydziergania jednego lub kilku oczek przypominającego warkocz łańcuszka (w zależności od rodzaju ściągów występujących w rzędzie), co nadaje nowemu rzędowi pożądaną wysokość. Dzięki tym dwóm zabiegom sposób wykonywania kwadratów szydełkowych



nie jest zupełnie spiralny. Ostateczny kwadratowy kształt robótki jest wynikiem dodania większej liczby oczek w rogach zewnętrznych rzędów.

Sposób zaczepiania nowych oczek w te z poprzedniego rzędu nie jest bez znaczenia dla wyglądu ostatecznego wyrobu. Wbijając szydełko i przeciągając nić przez środek oczka albo przez wolne przestrzenie wokół niego uzyskuje się płaskie lub bardziej przestrzenne wzory.



(a) Tworzenie łańcuszka z włóczki

(b) Zamykanie i tworzenie następnego rzędu

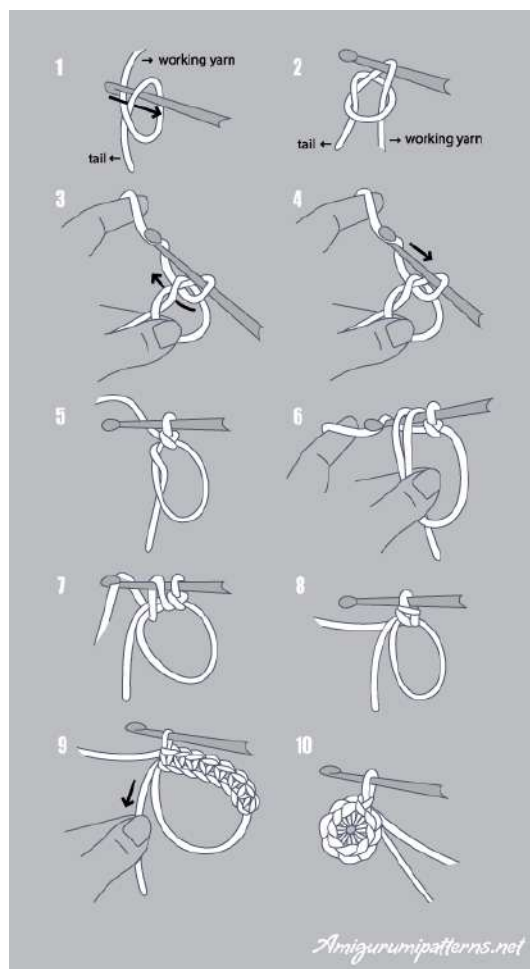
**Rysunek 3.2.** Technika pracy szydełkiem – łańcuszek (*Crochet chain stitch*, 2022)

Wspomniany łańcuszek to najprostszy, podstawowy wyrób szydełkowy. Sposób jego wykonywania przedstawiono na rysunku 3.2a. Nawet najbardziej skomplikowane wyroby szydełkowe często opierają się o jego wykorzystanie w różnych konfiguracjach. Kilka oczek łańcuszka zamkniętych w okrąg staje się początkiem (środkiem) kwadratu szydełkowego. Oczka następnego rzędu zaczepia się w kolejne pętelki łańcuszka lub w powstałą po jego zamknięciu przestrzeń. Drugi z przypadków widoczny jest na rysunku 3.2b, gdzie w kolistej luce wykonano szereg najprostszych ściegów szydełkowych – półsłupków. Łańcuszek służy nie tylko do rozpoczynania robótki i nowych rzędów po zamknięciu poprzednich, ale także do wyrabiania koronkowych wzorów i powiększania zewnętrznych rzędów w rogach, co nadaje kwadratowi ich właściwy kształt.

Początek kwadratu szydełkowego można wykonywać także w inny sposób, za pomocą zaciskowej pętli (ang. *magic ring*) widocznej na rysunku 3.3. Najwygodniej jest ją zawiązać za pomocą szydełka wokół palców dłoni. Powstaje wtedy luźny węzeł, wewnątrz którego można zaczepić dowolną liczbę oczek, a następnie zacisnąć, aby zredukować wolną przestrzeń lub rozciągnąć, aby dodać kolejne oczka. Umożliwia to tworzenie ściślejszych środków niż z użyciem zamkniętego łańcuszka.

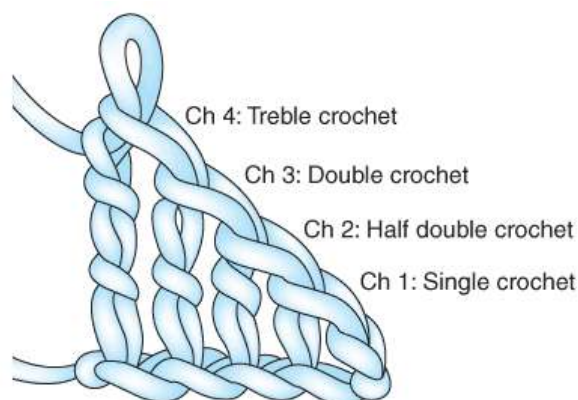
Rysunek 3.4 przedstawia podstawowe ściegi szydełkowe, zaczepione w środek pętelek łańcuszka. Różnią się one przede wszystkim wysokością. Im wyższy ścieg, tym szybciej zwiększa się średnica robótki. W jednym oczku łańcuszka można wydziierać np. kilka słupków lub kilka słupków utworzonych w różnych oczkach połączyć ze sobą od góry, tworząc pęczki. Nawet takie proste operacje pozwalają na tworzenie skomplikowanych

### 3. Technika szydełkowania i dostępne dane



**Rysunek 3.3.** Technika pracy szydełkiem – pętelka zaciskowa (*Magic ring crochet*, 2022)

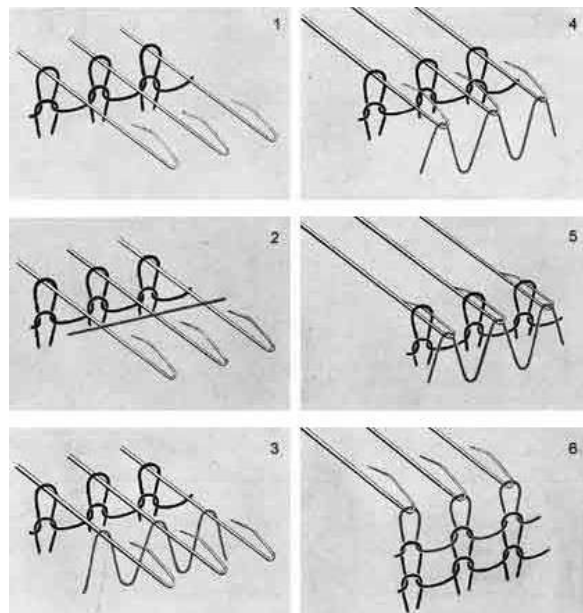
wzorów. Większość spotykanych kwadratów szydełkowych to złożenie prostych ściegów uzupełnionych o nadające robótce lekkości i delikatności łańcuszki.



**Rysunek 3.4.** Różne ściegi szydełkowe. Od prawej: oczko ścisłe (nieoznaczone), półsłupek (1), półsłupek nawijany (2), słupek pojedynczy (3), słupek podwójny (4) (*Lesson 10: How to Crochet Turning Chain*, brak daty)

Szydełkowanie zasadniczo różni się od z pozoru podobnego robienia na drutach. Ten drugi rodzaj dziewiarstwa polega na wykorzystaniu dwóch ostro zakończonych drutów

w przeciwieństwie do jednego, haczykowego szydełka. W przypadku pracy ręcznej dziewiarz ma na drutach nałożoną określoną liczbę pętli, która decyduje o szerokości końcowego wyrobu. Ze względu na zasadę wykonywania, czynność tę można efektywnie zmechanizować, pracuje się bowiem w jednej płaszczyźnie. Kolejne rzędy najprostszej, płaskiej dzianiny tworzy się poprzez przeciąganie nitki i przekładanie nowych utworzonych pętelek z jednego drutu na drugi, aż do końca rzędu. Następnie całą robótkę odwraca się na drugą stronę i powtarza czynność. W procesie maszynowym pętelki rozpięte są na szeregu haczyków. Maszyna wykonuje podobne operacje jak dziergająca osoba, ale zamiast trzymać pętelki na drutach, trzyma każdą z nich na osobnym haczyku. Nowe pętle tworzy równoległe na wielu haczykach, dzięki czemu jest w stanie wykonać cały nowy rząd pętelek na raz w trakcie jednego cyklu. Na rysunku 3.5 widać, że otrzymana dzianina (kafelek nr 6) przypomina połączone ze sobą bokami łańcuszki szydełkowe.

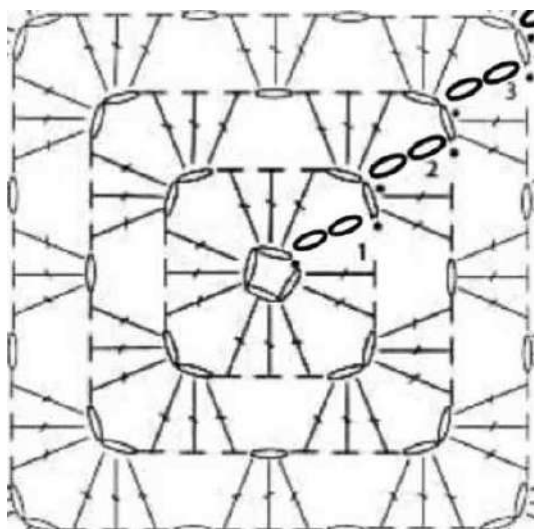


**Rysunek 3.5.** Cykl pracy maszyny dziewiarskiej (Ryj, 2008)

Za pomocą szydełka można imitować dzianinę robioną na drutach, wykorzystując wyłącznie oczka ściśle mimo odmienności techniki. Warto zaznaczyć, że w przypadku szydełkowania aktywna jest tylko jedna bieżąca pętela na raz. Każde oczko wykonuje się bowiem niezależnie od pozostałych, a po jego wykonaniu narzędzie można wyciągnąć z robótki bez obawy o utratę jej formy. Dzianina zsunięta nagle z drutów łatwo się rozplata, ponieważ pętelki z ostatniego rzędu są zależne bezpośrednio tylko od tych z poprzedniego, podczas gdy w dzianinie szydełkowej oczka podtrzymywane są także przez oczka po bokach. Sposób nałożenia pętelek na druty sprawia, że można na nich wykonywać w zasadzie jedynie prostą, prostokątną tkaninę. Wyroby okrągłe i przestrzenne wymagają użycia większej liczby drutów naraz.

Robótki szydełkowe wykonuje się na podstawie instrukcji słownych, które w następujących po sobie poleceniach niosą informacje o sposobie wykonania, tzn. nazwie i liczbie oczek oraz sposobie łączenia. Powtórzenia pewnych operacji zaznacza się, tworząc zapis przypominający mnożenie zawartości nawiasu. Do wad podejścia tekstowego należy

### 3. Technika szydełkowania i dostępne dane



(a) Schemat graficzny

Start with a magic ring:

1. Round 1: ch 4 (count as a "dc" the first 2 ch), \* 3 dc, 2 ch\*, repeat \*-\* 2 more times; 2 dc. Sl. st to close the round, sl. st in next ch space.

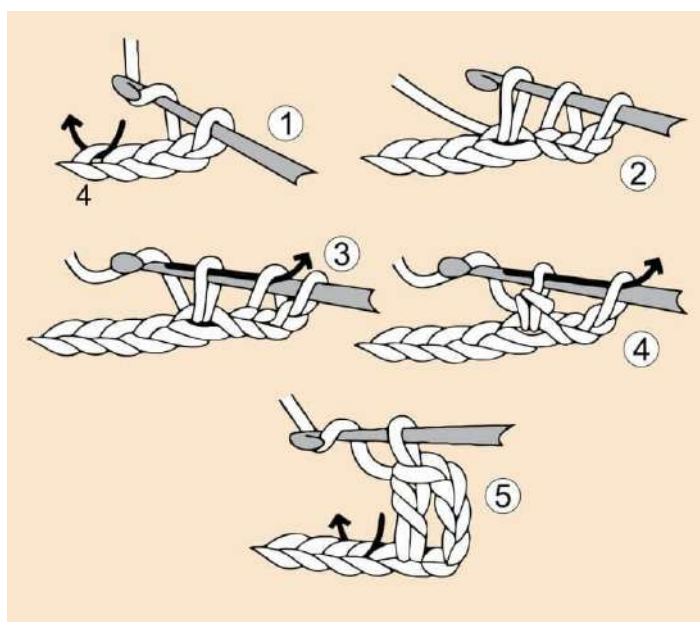
2. Round 2: ch 4 (count as a "dc" the first 2 ch); \*3 dc in the same ch space, 1 ch, 3 dc in next ch space, 2 ch\* repeat \*-\* 2 more times; 3 dc in the same ch space, 1 ch, 2 dc in next ch space. Sl. st to close the round, sl. st in next ch space.

3. Round 3: ch 4 (count as a "dc" the first 2 ch); \*3 dc in the same ch space, 1 ch, 3 dc in next ch space, 1 ch, 3 dc in next ch space, 2 ch\*; repeat \*-\* 2 more times; 3 dc in the same ch space, 1 ch, 3 dc in next ch space, 1 ch, 2 dc in next ch space. Sl. st to close the round, sl. st in next ch space.

(b) Instrukcje słowne

**Rysunek 3.6.** Instrukcje do wykonania prostego kwadratu szydełkowego (10 crochet granny square, 2021)

przede wszystkim nieczytelność przy większej objętości tekstu. Ponadto następujące po sobie instrukcje nie dają bezpośredniego wyobrażenia o końcowym wyrobie, przez co łatwiej o pomyłkę.



**Rysunek 3.7.** Sposób wykonywania słupek pojedynczego (US Crochet Stitch Terms, 2018)

Problemem związanym z nazewnictwem, a w efekcie z instrukcjami słownymi, jest także różne postrzeganie tych samych ściągów w różnych językach, np. słupek pojedynczego przedstawionego na rysunku 3.7. Odpowiednik w amerykańskiej odmianie języka angielskiego to *double crochet*. Jeszcze mniej intuicyjne jest określenie na ten rodzaj ścięgu w brytyjskiej odmianie języka – *treble crochet*. Analizując tabelę 3.1, warto zwrócić uwagę, że nazwa *double crochet* znaczy co innego w obu odmianach języka angielskiego, co bez

znajomości kraju pochodzenia schematu może utrudnić prawidłową interpretację instrukcji.

**Tabela 3.1.** Porównanie nazewnictwa podstawowych ściegów szydełkowych w języku polskim oraz w amerykańskiej i brytyjskiej odmianie języka angielskiego (wraz ze skrótami) (*Jak szydełkować po amerykańsku*, 2015)

Nazwa polska	Nazwa amerykańska	Nazwa brytyjska
łańcuszek	chain (ch)	chain (ch)
półsłupek	single crochet (sc)	double crochet (dc)
półsłupek nawijany	half double crochet (hdc)	half treble crochet (htr)
słupek pojedynczy	double crochet (dc)	treble crochet (tr)
słupek podwójny	treble/triple crochet (trc)	double treble crochet (dtr)
oczko ścisłe	slip stitch (sl st/ss)	slip stitch (sl st/ss)

Ze względu na duży poziom złożoności nawet prostych robótek dla skrócenia instrukcji stosuje się różne skróty nazw ściegów z ograniczonego słownika. Tym sposobem „2 półsłupki nawijane łączone górą” skraca się do „ps.naw2ł”. Amerykańską nazwę *half double crochet decrease* można skrócić do *hdc2tog*. Taki zabieg co prawda zmniejsza długość instrukcji, ale nie wpływa korzystnie na jej czytelność. Nie każdy autor decyduje się jednak na taki krok, stawiając za to bardziej na polecenia opisowe, przez co instrukcje słowne nie podlegają żadnym ogólnym zasadom i normom.

W porównaniu do instrukcji słownych schematy obrazkowe mają istotną zaletę, która sprawia, że są preferowane przez wielu dziewiarzy. Swoim kształtem przypominają uproszczoną wersję gotowego wyrobu, a więc już w trakcie pracy korzystający z nich może na bieżąco kontrolować efekty swojej pracy i wcześniej wykrywać pomyłki. Na podstawie instrukcji graficznych można także oszacować ostateczny rozmiar wyrobu, mimo że jest to wypadkowa także grubości nici i rozmiaru użytego szydełka. Schematy nie są jednak wolne od wad – nawet w przedstawieniach graficznych spotykane są pewne różnice w notacji, o czym więcej w kolejnym rozdziale. Nie stanowią one jednak większej przeszkody dla doświadczonego dziewiarza.

### 3.2. Dostępne dane

Niniejsza praca koncentruje się wokół graficznych schematów kwadratów szydełkowych. Wybór wzorów obrazkowych jako danych uczących dla sztucznej sieci neuronowej umożliwia łatwiejszą analizę wyników algorytmu uczenia maszynowego, ponieważ co do zasady wzory te mają przypominać gotowy wyrób szydełkowy. Ocena przydatności instrukcji słownych ze względu na ich skomplikowanie i brak standardów byłaby trudna, ponieważ nie istnieją żadne jednoznaczne wyznaczniki ich jakości i przydatności. Faktyczną użyteczność można byłoby ocenić, przeprowadzając próbę wykonania na ich podstawie robótki, co byłoby bardzo czasochłonne.

Schematy obrazkowe kodują pewne informacje za pomocą znaków ze złożonego alfabetu. Ze względu na region pochodzenia autora, tworzone przez niego schematy mogą różnić się stosowaną notacją, co może dezorientować niedoświadczonego odbiorcę. Wpra-



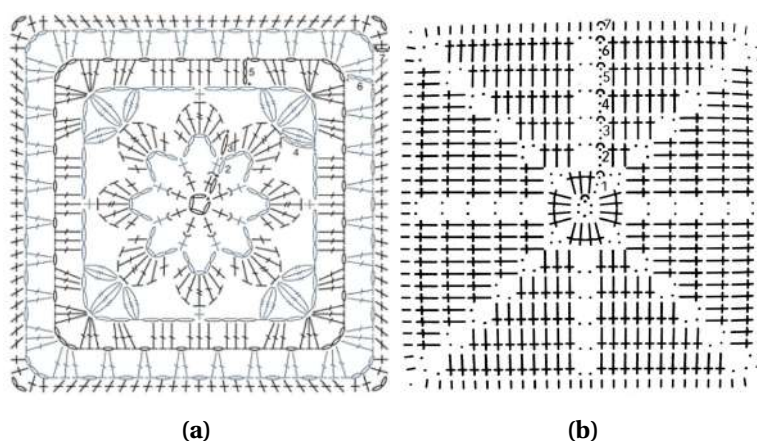
### 3. Technika szydełkowania i dostępne dane

wiony dziewiarz wie jednak, że cienkie, łukowate skupiska to łańcuszki, bez względu na to, czy oznaczono je za pomocą okręgów, elips czy kropek. Praktyka pozwala mu spodziewać się zamknięć rzędów w określonych miejscach, po prawej stronie od łańcuszka rozpoczynającego ten sam rząd (po wykonaniu szeregu ściągów odwrotnie do ruchu wskazówek zegara). Tam zazwyczaj pojawia się znak oczka ścisłego w postaci kropki lub łuku nad oczkiem, w którego miejscu należy utworzyć zamknięcie. Na niektórych schematach takie oznaczenie nie występuje jednak wcale, mimo że dla zachowania właściwego kształtu robótki rzędy należy zamykać (prawie) zawsze.

**Tabela 3.2.** Najczęściej spotykane oznaczenia graficzne podstawowych ściągów szydełkowych

Nazwa ściegu	Spotykane oznaczenia
łańcuszek	okrąg, elipsa, kropka
oczko ścisłe	kropka, łuk (oczko zamykające)
półsłupek	x, + lub kreska (czasem kreślona)
półsłupek nawijany	T
słupek pojedynczy	T kreślone lub kreska pionowa kreślona
słupek podwójny	T podwójnie kreślone lub kreska pionowa podwójnie kreślona
słupek potrójny	T potrójnie kreślone lub kreska pionowa potrójnie kreślona

Tabela 3.2 to zbiór najczęściej spotykanych oznaczeń graficznych podstawowych ściągów szydełkowych. Zazwyczaj półsłupki nawijane i różne rodzaje słupków przedstawia się za pomocą znaków przypominających dużą literę T, która w przypadku słupków jest kreślona jedno- lub wielokrotnie w zależności od kodowanego ich rodzaju. Na tym samym schemacie półsłupek oznacza się wówczas dla czytelności jako małą lub dużą literę X, znak dodawania (+), pionową kreskę, czy też pionową kreskę kreśloną ukośnie, przypominającą krzyż z pochyłą poprzeczką. Taką notację zastosowano na rysunku 3.8a. Na pojedynczy płatek kwiatka składa się kilka rodzajów ściągów wbitych w łańcuszek: półsłupek oznaczony jako krzyżyk z pochyłą poprzeczką, półsłupek nawijany jako litera T, słupek jako litera T kreślona, słupek podwójny jako litera T kreślona podwójnie.

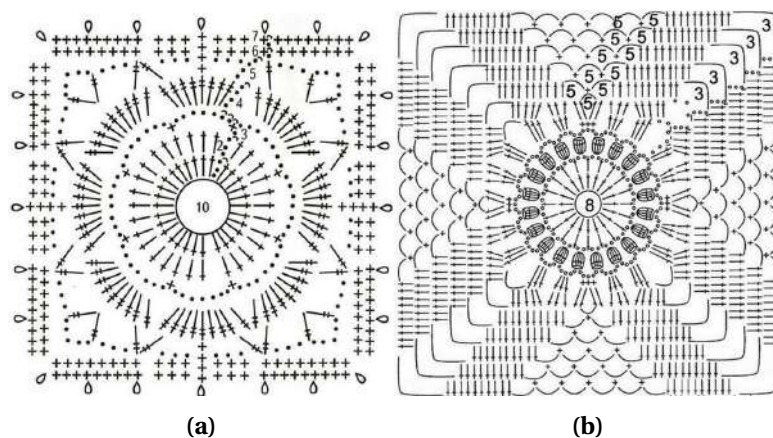


**Rysunek 3.8.** Przykłady schematów z różną notacją słupków i półsłupków

Są także schematy, gdzie słupki oznaczają się za pomocą znaków krzyża, a nie za pomocą dużej litery T kreślonej. Rysunek 3.8b przedstawia przykład tej odmiennej notacji. Półsłupki na obwódce – dla odróżnienia – oznaczono krótkimi kreskami.

Warto także zwrócić uwagę, że grafiki 3.8a i 3.8b różnią się sposobem przedstawienia łańcuszków. Oba wzory tylko schematycznie przedstawiają kwadrat szydełkowy, ale 3.8b robi to w sposób znacznie bardziej uproszczony. Zazwyczaj, jeżeli słupki są kodowane na kształt litery T, to łańcuszek jest rysowany jako elipsa, a jeżeli są oznaczane jako krzyże, to łańcuszki są upraszczane do kółeczek lub wręcz do małych kropek.

Podobnie jak instrukcje słowne, schematy obrazkowe także zawierają pewne uproszczenia zwiększające przejrzystość. Obejmują one najczęściej fragmenty z łańcuszkami – na grafice 3.9a tylko początek robótki w środku, a na 3.9b wszystkie łańcuszki zastąpiono uproszczonymi łukami.

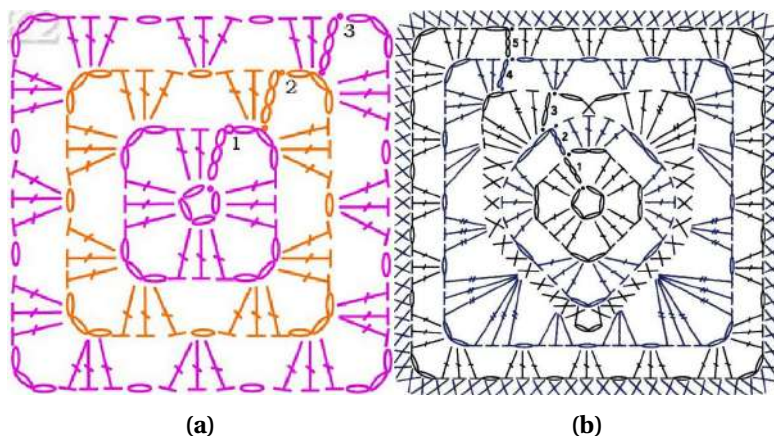


**Rysunek 3.9.** Przykładowe sposoby przedstawienia początku schematu szydełkowego w postaci łańcuszka

Podczas analizy grafik 3.9 należy zwrócić uwagę na obecne na nich adnotacje. Dodatkowe oznaczenia w postaci liczb towarzyszą uproszczeniom i mówią o liczbie oczek koniecznych do powtórzenia. Na schematach zdarzają się także etykiety numerujące kolejne rzędy.

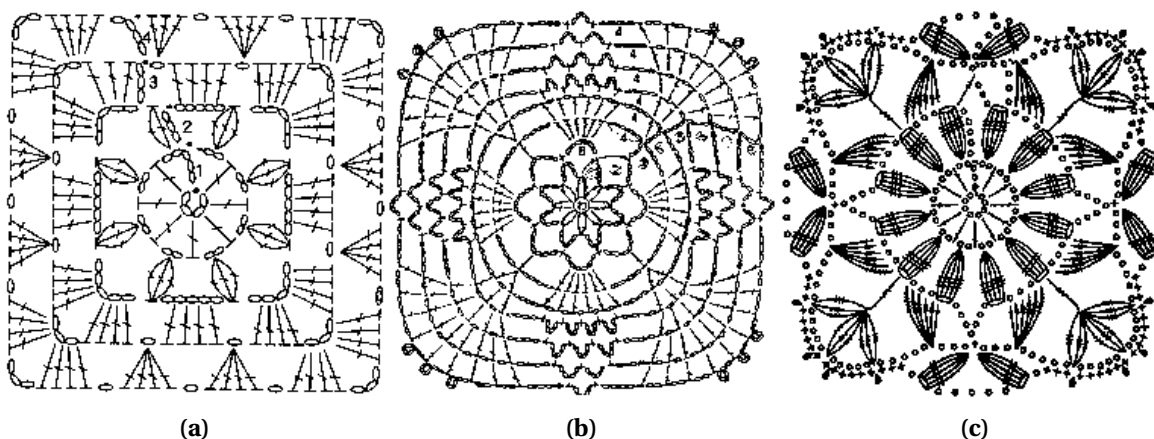
Ponieważ nie istnieją publikacje związane z wykorzystaniem sztucznej inteligencji do generowania obrazkowych schematów szydełkowych, brakuje także zawierających je zbiorów danych. Na potrzeby niniejszej pracy zebrano wzory szydełkowe rozproszone w Internecie. Przedstawiają one niewielkie robótki, przeważnie w formie kwadratu. Dołączono także formy okrągłe i wieloboczne, ponieważ mimo nazwy podlegają pod termin *granny square*. Wyeliminowano grafiki niskiej jakości, które były nieczytelne dla człowieka. Pozostawiono część duplikatów wzorów, jako że nie były to kopie plików obrazów, a różne pliki przedstawiające taki sam schemat szydełkowy. Zbiór danych składa się z 2554 grafik, które podzielono na dwa podzbiory: jeden ze wzorami symetrycznymi, z czterema osiami symetrii (2364 egzemplarze), a drugi z niesymetrycznymi (190 egzemplarzy). Do uczenia maszynowego wykorzystano podzbiór grafik symetrycznych.

Nie wszystkie zebrane obrazy przedstawiały sam wzór szydełkowy. Bardzo wiele z nich było połączonych także ze zdjęciem gotowego wyrobu czy z dodatkowym opisem. Podjęto



**Rysunek 3.10.** Przykład wzoru symetrycznego (z czterema osiami symetrii) i niesymetrycznego

próbę filtracji pożądanych informacji z użyciem prostego algorytmu wyznaczającego entropię natężenia koloru w kanałach RGB. Algorytm ten sprawdzał się tylko w niektórych przypadkach, dlatego ostatecznie obrazy przycięto ręcznie za pomocą prostego programu graficznego. Następnie odfiltrowano część najmniejszych grafik i zmniejszono rozmiar pozostałych do 256 x 256 pikseli. Obrazy przetworzono do skali szarości, ponieważ różnorodne kolory nie niosły informacji istotnej z punktu widzenia uczenia maszynowego, a służyły tylko niewielkiemu zwiększeniu czytelności wzorów dla człowieka. Dzięki temu zmniejszono także późniejszy nakład obliczeniowy.



**Rysunek 3.11.** Przykłady zbinaryzowanych wzorów ze zbioru danych

Kolejno wyszarzone obrazy zbinaryzowano za pomocą biblioteki OpenCV. Ze względu na różnorodność zebranych danych, także różny poziom ich szumu, użyto metod automatycznie progujących obraz. Przy wzorach, w których rzędy oczek po wyszarzeniu nie różniły się zbyt wiele między sobą natężeniem koloru, najlepiej sprawdziła się metoda Otsu. Ten sam algorytm jednak wymazywał niekiedy całe rzędy w pewnym ułamku grafik, jeżeli np. w oryginalnym obrazie niektóre rzędy oczek były zaznaczone na jasnożółto. Dla tego przypadku najlepiej sprawdził się algorytm adaptacyjny, gdzie próg stanowi suma wartości w sąsiedztwie z wagami wyznaczanymi z rozkładu Gaussa.



## 4. Zaproponowane rozwiązanie

### 4.1. Przegląd literatury

Istnieją jedynie nieliczne publikacje o związku dziewiarstwa ze sztuczną inteligencją, a większość z nich ma charakter amatorski. Autorka Janelle Shane w ramach projektu *SkyKnit* wykorzystwała rekurencyjną sieć neuronową do wygenerowania instrukcji słownych do robienia na drutach. Model został wytrenowany na zbiorze składającym się z 5228 instrukcji. Niestety sieć nawet w najlepszych przypadkach powtarzała rzędy, gubiła oczka i generowała niemożliwe lub trudne do wykonania ich kombinacje (Madrigal, 2018).

Następcą *SkyKnit* był projekt *HAT3000* tej samej autorki, który ograniczał się do tworzenia instrukcji szydełkowych do wydziergania czapek. Shane wybrała model Open AI GPT-2-355M z dłuższą pamięcią i poduczony na różnych danych z Internetu, niezwiązanych z szydełkowaniem. Czapkę szydełkową w klasyczny sposób wykonuje się od czubka głowy, naddając w kolejnych rzędach określoną liczbę oczek, aby zwiększyć obwód. Generowane przez sieć wzory zaczynały się poprawnie, ale w następnych instrukcjach pojawiały się niewłaściwe liczby naddań, przez co liczba oczek w kolejnych rzędach rosła coraz szybciej. Czapki wykonane na podstawie wzorów wygenerowanych przez *HAT3000* przypominały hiperboliczne superpowierzchnie (Shane, 2019).

Naukowcy z Massachusetts Institute of Technology w 2019 r. ogłosili projekt, w ramach którego opracowali metodę automatycznego generowania instrukcji dziewiarskich na podstawie zdjęć dzianiny. W jego ramach przygotowali zbiór danych składający się ze zdjęć rzeczywistych próbek wraz z ich odpowiednikami w formie instrukcji graficznych. W artykule „*Neural Inverse Knitting: From Images to Manufacturing Instructions*” (Kaspar, Oh i in., 2019) zaprezentowali zastosowanie generatywnych sieci antagonistycznych (ang. GAN, *Generative Adversarial Networks*) z warstwami splotowymi do generowania instrukcji dziewiarskich na podstawie zdjęć gotowej dzianiny. W powiązanej publikacji „*Knitting Skeletons: A Computer-Aided Design Tool for Shaping and Patterning of Knitted Garments*” (Kaspar, Makatura i Matusik, 2019) ta sama grupa naukowców przedstawiła oprogramowanie pozwalające tworzyć wzory dla maszyn dziewiarskich. Obsługa takich urządzeń wymaga nieco doświadczenia, wobec tego program miałby uprościć przebieg przygotowań do prac, przerzucając część odpowiedzialności z człowieka na złożone algorytmy uczenia maszynowego.

Dostępna literatura porusza przede wszystkim związek uczenia maszynowego z robieniem na drutach. Jedynie projekt *HAT3000* dotyczy możliwości tworzenia instrukcji szydełkowych, lecz są to instrukcje słowne. W związku z tym, że niniejsza praca koncentruje się wokół graficznych schematów szydełkowych, wymienione projekty stanowiły jedynie źródło inspiracji.

### 4.2. Metodyka

Współczesne sztuczne sieci neuronowe są modelami wysokiej klasy, które pozwalają użytkownikowi na uzyskanie zadowalających wyników na podstawie nieprzetworzonych

danych wejściowych. Modele ogólnego przeznaczenia nie zawsze są wystarczające w wąskich zastosowaniach. Z innej strony przygotowanie specjalistycznych architektur wiąże się z dodatkowymi wyzwaniami. Wobec tego bardzo często wykorzystanie gotowych struktur sieci neuronowych lub gotowych poduczonych modeli poprzedza się przetwarzaniem wstępnym.

Ze względu na złożoność zagadnienia poruszanego w niniejszej pracy jako model sztucznej sieci neuronowej spośród modeli generatywnych wybrano autoenkoder z wymuszeniem wariancji. Rozważano także inne architektury, jak generatywne sieci przeciwstawne, które ostatecznie odrzucono. Sieci te są trudniejsze w implementacji i dostrajaniu, wymagają także więcej próbek uczących. Ponadto przeprowadzono kilka prób z rekurencyjną siecią DRAW (Gregor i in., 2015). Jej architektura składa się z sieci enkoder-dekoder i wykorzystuje mechanizmy uwagi do iteratywnego generowania obrazów. Ostatecznie nie kontynuowano pracy z tą siecią.

Ponieważ brakuje kryteriów oceny jakości instrukcji szydełkowych, w trakcie analizy wyników zwracano uwagę na zaszumienie obrazu, a wartość otrzymanych schematów wyceniano na podstawie złożoności otrzymanych wzorów.

### 4.3. Kodowanie danych

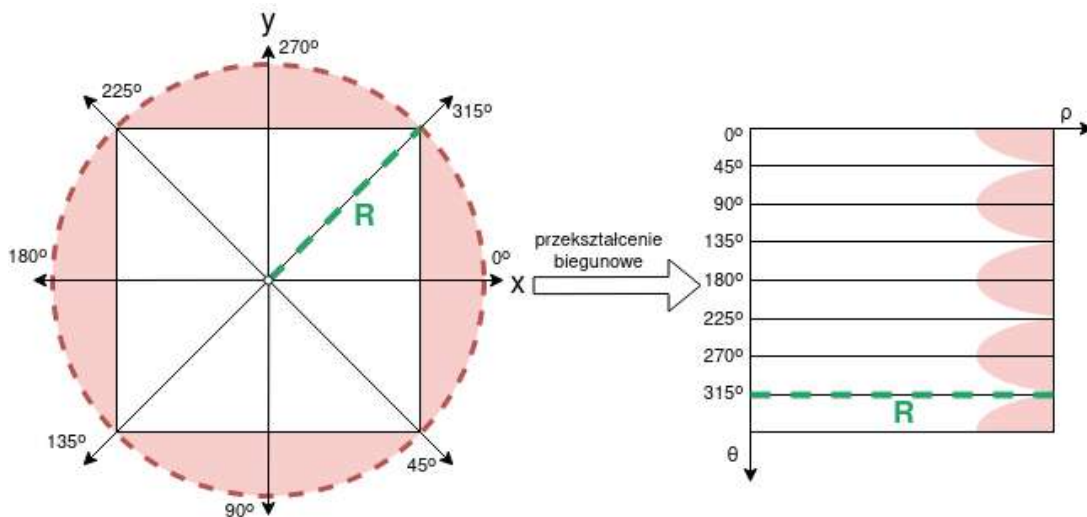
Z natury kwadratów szydełkowych wynika, że każda taka robótka rozpoczyna się od kolistego początku, a kolejne rzędy są dodawane wzdłuż obwodów poprzednich. Powstają wówczas symetryczne wzory, które na zewnątrz układają się w kształt kwadratu. Zauważając tę właściwość i stosując analogię do kodowania symetrycznych, okrągłych obiektów w układzie współrzędnych biegunowych, można założyć, że kodowanie w tym układzie współrzędnych przyniesie korzyści z punktu widzenia uczenia wybranego modelu. Właśnie takie charakterystyczne cechy badanych struktur wykorzystywane są w obszarze medycyny, gdzie koliste obiekty koduje się niekiedy w dwóch układach współrzędnych: kartezjańskim i biegunowym. Czerpanie informacji tylko z pierwszego z nich nie zawsze jest wystarczające. Ponieważ kształt i układ obiektów jest zupełnie różny w układzie kartezjańskim i biegunowym, inne, komplementarne informacje zostają wyciągnięte z obu układów, co wzbogaca reprezentację cech (Liu i in., 2019).

Należy zaznaczyć, że algorytm wykonujący przekształcenie obrazu z kartezjańskiego do biegunowego układu współrzędnych zazwyczaj nie jest częścią sieci neuronowej. Przekształcenie odbywa się przed „wejściem” do modelu (Benčević i in., 2021; Cheng i in., 2020; B.-S. Kim i in., 2018; J. Kim i in., 2020; Liu i in., 2019). Niektóre źródła wskazują jednak, że takie podejście może nie być wystarczające i proponują uczyć się moduł transformacji biegunowej, który jest już częścią sieci (Feng i in., 2022).

Aby przeprowadzić poprawne przekształcenie schematów z jednego układu współrzędnych do drugiego, należy zlokalizować środek obrazka w układzie kartezjańskim, który stanie się początkiem biegunowego układu współrzędnych (J. Kim i in., 2020). Literatura opisuje różne metody wyznaczania środka, a wybór podejścia powinien być podyktowany wynikami odpowiednich rozważań co do przeznaczenia sieci. Jeżeli przewiduje się, że tworzona sieć będzie przetwarzać nowe dane bez etykiet, warto wybrać metodę, która będzie

pozwalą na automatyczne wyznaczenie środka na podstawie obrazu (Benčević i in., 2021). Takie podejście wiąże się jednak z dodatkowymi nakładami wynikającymi z konieczności przygotowania algorytmu do tego specyficznego celu. W związku z tym bardzo często dla ułatwienia jako początek biegunowego układu współrzędnych obiera się po prostu geometryczny środek obrazu wejściowego (J. Kim i in., 2020).

Zbiór danych ze schematami szydełkowymi został częściowo opracowany ręcznie – obrazy przycięto za pomocą programu graficznego. Taka operacja oczywiście nie jest stuprocentowo precyzyjna, tym bardziej że same schematy to często nieidealne rysunki tworzone ręcznie. W rezultacie środki schematów mogą być odsunięte o kilka pikseli od rzeczywistego geometrycznego środka obrazu. Z tych powodów początki biegunowego układu współrzędnych wskazano ręcznie. Metoda ta była optymalna do tego celu – lepsza niż najprostsze geometryczne przybliżenie, a wciąż stosunkowo mało czasochłonna przy odpowiednim skonfigurowaniu programu graficznego. Ponieważ wszystkie elementy zbioru danych zostały wcześniej zbinaryzowane (wartości pikseli 0 i 255), do zaznaczenia środków wykorzystano pojedynczy szary piksel (wartość 130) naniesiony ręcznie w programie graficznym GIMP.



**Rysunek 4.1.** Przekształcenie biegunowe w bibliotece OpenCV

Celem jest przekształcenie obrazu wyjściowego w kształcie kwadratu z układu współrzędnych kartezjańskich do biegunowych z zachowaniem rozmiaru w pikselach. Biblioteka OpenCV przeprowadza transformację biegunową postaci  $dst(\rho, \theta) = src(x, y)$  widoczną na rysunku 4.1 zgodnie z poniższymi zależnościami (OpenCV, 2015):

$$\theta = \frac{H}{2\pi} \cdot \text{angle}(x - x_0, y - y_0), \quad (4.1)$$

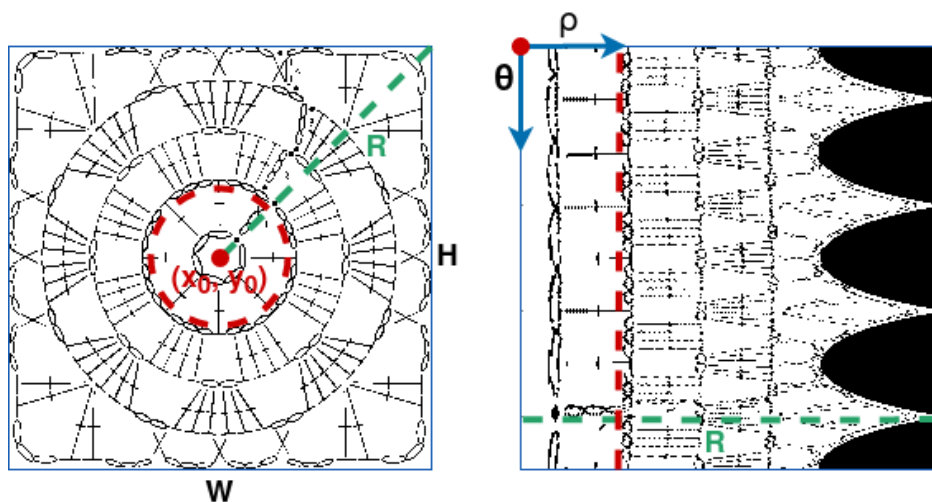
$$\rho = \frac{W}{R} \cdot \text{magnitude}(x - x_0, y - y_0), \quad (4.2)$$

gdzie:

$$\text{magnitude}(x, y) = \sqrt{x^2 + y^2}, \quad (4.3)$$

$$\text{angle}(x, y) = \text{atan2}(y, x), \quad (4.4)$$

przy czym  $\text{atan2}$  to dwuargumentowa funkcja arcus tangens. Aby objąć przekształceniem całą istotną informację, należy wpisać kwadrat w okrąg. Ponieważ promień tego okręgu jest dłuższy niż połowa długości kwadratu, przekształcenie obejmuje także obszar poza nim, w którym nie ma informacji. Biblioteka rozwiązuje ten problem, dopełniając tensor czarnymi pikselami. Na rysunku 4.1 odpowiednie obszary zaznaczono na czerwono. W efekcie, który widać na rys. 4.2, powstaje obraz z czarnymi, parabolicznymi obszarami.



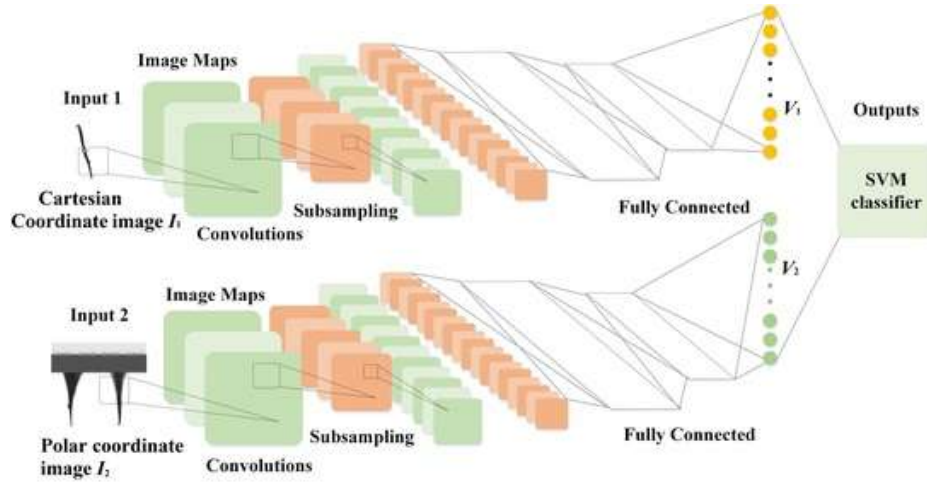
**Rysunek 4.2.** Zależność pomiędzy kartezjańskim i biegunowym układem współrzędnych

Sieci neuronowe kodujące w biegunowym układzie współrzędnych znajdują zastosowanie w medycynie, ponieważ wiele struktur w ludzkim ciele, jak np. tarcza nerwu wzrokowego, ma kształt okrągły lub eliptyczny. Operowanie na tym układzie poprawia rozróżnienie pikseli obiektu od pikseli tła (Liu i in., 2019). Ponieważ w niniejszej pracy czarne schematy dobrze odróżniają się od białego tła, użycie biegunowego układu współrzędnych wzbogaca proces wyciągania informacji o zasadach tworzenia ściegów i rzędów w robótkach szydełkowych, począwszy od okrągłych środków aż po kwadratowe brzegi.

Sporadycznie pojawiają się także publikacje naukowe przedstawiające „dwugałęziowe” architektury sieci neuronowych, gdzie jedna gałąź koduje obrazy w układzie kartezjańskim, a druga (równoległe) w biegunowym. Gałęzie można trenować niezależnie, bez krzyżowej wymiany informacji, a dopiero na końcu pobierać odpowiednie informacje z obu z nich i łączyć je przed dalszymi operacjami (Cheng i in., 2020), np. klasyfikacją, jak na rysunku 4.3. Spotykane są także rozwiązania implementujące specjalne moduły łączące zapewniające zgodność pomiędzy dwoma układami współrzędnych w czasie uczenia sieci neuronowej. Dzięki temu model uczy się komplementarnych informacji o kontekście (Liu i in., 2019).

#### 4.4. Sieć typu autoenkoder wariacyjny

Niegdyś używano autoenkoderów do redukcji wymiaru (ang. *dimensionality reduction*) lub wyuczania się cech (ang. *feature learning*), ale w ostatnich latach znalazły one także zastosowanie w modelowaniu generatywnym. Zawdzięcza się to powiązaniom matema-



**Rysunek 4.3.** Przykład dwugłęziowej architektury sieci neuronowej do zadania klasyfikacji (Cheng i in., 2020)

tycznym zasad działania tych sieci z zasadami dotyczącymi modeli ze zmiennymi ukrytymi (ang. *latent variable models*) (Goodfellow, Bengio i Courville, 2016).

Przyjmuje się, że dla każdej próbki  $x$  w zbiorze danych  $X$  istnieje taka zmienna ukryta  $z$  (lub ich zestaw), która sprawia, że model generuje wynik podobny do  $x$ . Oznacza to, że możliwe jest próbkowanie wektora zmiennych ukrytych z wielowymiarowej przestrzeni  $\mathcal{Z}$  zgodnie z pewną funkcją gęstości prawdopodobieństwa  $p_\theta(z)$ . Działanie modeli generatywnych bazuje zatem na szukaniu pewnej funkcji gęstości prawdopodobieństwa  $p_\theta(x)$ , która maksymalizuje prawdopodobieństwo wystąpienia zmiennej obserwowanej  $x \in X$ .

Najczęściej obieranym kryterium w tego typu modelach jest metoda maksymalnej wiarygodności

$$p_\theta(x) = \int p_\theta(x, z) dz, \quad (4.5)$$

gdzie  $p_\theta(x, z) = p_\theta(z)p_\theta(x|z)$ . Rozkłady  $p_\theta(z)$  i  $p_\theta(x|z)$  są znane;  $p_\theta(z)$  określa się jako rozkład a priori, ponieważ nie zależy od żadnych obserwacji (Kingma, Welling i in., 2019). Obliczenie przedstawionej powyżej całki wiąże się z koniecznością wyznaczenia prawdopodobieństwa krańcowego (Biesiada, 2014), wobec tego jest ona niewyznaczalna (gęstość a posteriori  $p_\theta(z|x) = p_\theta(x|z)p_\theta(z)/p_\theta(x)$  jest niewyznaczalna). Dla rozkładu  $p_\theta(z|x)$  wprowadza się zatem przybliżenie postaci  $q_\phi(z|x)$  (Kingma i Welling, 2013). Dla dowolnego wybranego modelu wnioskowania  $q_\phi(z|x)$  (włącznie z parametrami  $\phi$ ) prawdą jest, że:

$$\log p_\theta(x) = \text{ELBO} + D_{KL}(q_\phi(z|x) || p_\theta(z|x)). \quad (4.6)$$

Drugi z członów w równaniu 4.6 to dywergencja Kullbacka-Leiblera, która wymusza podobieństwo między rozkładami  $q_\phi(z|x)$  oraz  $p_\theta(z|x)$  (Kingma, Welling i in., 2019). Ponieważ jest ona nieujemna, pierwszy człon równania, ELBO, stanowi dolne ograniczenie wariacyjne (ang. *evidence lower bound, variational lower bound*).

Autoenkoder wariacyjny składa się z dwóch sieci neuronowych: enkodera i dekodera. Pierwsza z nich dostarcza właśnie przybliżony rozkład a posteriori  $q_\phi(z|x)$  charakteryzo-

#### 4. Zaproponowane rozwiązanie

---

wany przez wagi sieci oznaczane zbiorowo jako  $\phi$ , a druga rzeczywisty rozkład  $p_\theta(x|z)$  charakteryzowany przez wagi  $\theta$  (Odaibo, 2019).

Z drugiego członu w równaniu 4.6 można wyprowadzić, że (Odaibo, 2019):

$$\log p(x_i) \geq -D_{KL}(q_\phi(z|x_i)||p_\theta(z)) + E_{\sim q_\phi(z|x_i)}[\log p_\theta(x_i|z)]. \quad (4.7)$$

Prawa strona powyższego równania to ELBO (ang. *Evidence Lower Bound*), czyli dolne ograniczenie wariacyjne. Jego maksymalizacja pośrednio maksymalizuje  $\log p(x_i)$ , co jest podstawą wnioskowania wariacyjnego, jako że maksymalizacja wprost jest niemożliwa (Odaibo, 2019). Warto zaznaczyć, że zamiast maksymalizować ELBO można minimalizować przeciwieństwo (-ELBO).

Aby zoptymalizować dolne ograniczenie wariacyjne za pomocą metody gradientu prostego, trzeba wyznaczyć gradienty względem wszystkich parametrów (Fabius i Van Amersfoort, 2014). Pomimo dowolności w wyborze postaci ukrytego rozkładu a priori  $p(z)$  (ang. *latent prior*) i przybliżonego rozkładu  $q_\phi(z|x_i)$  zazwyczaj wybiera się rozkład normalny standardowy, tak że  $p(z) = N(0, I)$  (tzn.  $\mu = 0, \sigma = 1$ ). Następnie  $z$  można próbkować z rozkładu a posteriori, ponieważ  $z \sim q_\phi(z|x) = N(\mu, \sigma)$ ; wówczas dekodery powinien być w stanie zrekonstruować  $\tilde{x}$  na podstawie  $z$ . Niestety operacja próbkowania ze względu na losowość nie jest różniczkowalna, co uniemożliwia propagację wsteczną błędów i w rezultacie uczenie modelu. W związku z tym Kingma i Welling (2013) zaproponowali reparametryzację (ang. *reparametrization trick*, dosł. „sztuczka z reparametryzacją”), która polega na losowym próbkowaniu zmiennej  $\epsilon$  ze standardowego rozkładu normalnego  $N(0, I)$ , a następnie przeskalowaniu tej zmiennej o odchylenie standardowe i przesunięcie o średnią, tak aby otrzymać zmienną  $z$ :

$$z = \mu + \sigma \odot \epsilon \quad (4.8)$$

W ten sposób losowa zmienna  $z$  staje się zmienną deterministyczną  $z = g(\epsilon, x)$  (Ulger, Yuksel i Yilmaz, 2021).

Efektorem powyższych założeń jest następująca, ostateczna postać funkcji celu dla autoenkodera wariacyjnego:

$$\mathcal{L} = - \sum_{j=1}^J \frac{1}{2} \left[ 1 + \log(\sigma_j^2) - \sigma_j^2 - \mu_j^2 \right] - \frac{1}{L} \sum_l E_{\sim q_\phi(z|x_i)} \left[ \log p(x_i|z^{(i,l)}) \right], \quad (4.9)$$

gdzie  $J$  to wymiarowość wektora ukrytego  $z$ , a  $L$  to liczba próbek uzyskanych za pomocą reparametryzacji. Ze względu na wybór rozkładów normalnych pierwszy człon w powyższym równaniu jest wynikiem całkowania analitycznego  $D_{KL}(q_\phi(z|x_i)||p_\theta(z))$ . Warto zaznaczyć, że pełni on funkcję regularyzatora dla parametrów  $\phi$ , wymuszającego podobieństwo przybliżonego rozkładu a posteriori do rozkładu a priori  $p_\theta(z)$ . Drugi człon równania to oczekiwany błąd rekonstrukcji  $E_{\sim q_\phi(z|x_i)} \left[ \log p(x_i|z^{(i,l)}) \right]$ , który nie może być obliczony analitycznie, zatem musi zostać oszacowany na pewnej próbce danych uczących (Kingma i Welling, 2013).

Podsumowując, uczenie autoenkodera wariacyjnego polega na szukaniu optymalnych parametrów sieci neuronowej  $(\phi^*, \theta^*)$ , które minimalizują  $\mathcal{L}$  (Kingma i Welling, 2013; Odaibo, 2019):

$$(\phi^*, \theta^*) = \arg \min_{(\phi, \theta)} \mathcal{L}(\phi, \theta) \quad (4.10)$$

W autoenkoderach wariacyjnych sieci neuronowe pełnią funkcję probabilistycznych enkoderów i dekoderów. Zazwyczaj enkoder ma formę enkodera gaussowskiego (jak w powyższych rozważaniach). W przypadku dekodera stanowiącego część modelu uczącego się na danych ciągłych przyjmuje się, że rozkład  $p_\theta(x|z)$  jest wielowymiarowym rozkładem normalnym, a dla danych binarnych wybiera się rozkład zero-jedynkowy (Kingma i Welling, 2013). Wówczas o takich dekodernach można mówić (odpowiednio): dekodek gaussowski i dekodek Bernoulliego.

W praktyce działanie autoenkodera wariacyjnego można sprowadzić do kilku kroków:

1. sieć przyjmuje wejście w postaci próbki  $x$  (obrazu) ze zbioru danych  $X$ ;
2. enkoder wyznacza wektor średnich i odchyłeń standardowych każdego obrazu;
3. przeprowadzana jest reparametryzacja celem wyznaczenia wektora cech ukrytych  $z$ ;
4. dekodek rekonstruuje obraz na podstawie  $p_\theta(x|z)$  (Ulger, Yuksel i Yilmaz, 2021).

Uproszczony zapis funkcji celu dla autodekoderu wariacyjnego (równanie 4.9) to:

$$\mathcal{L} = -\text{ELBO} = \mathcal{L}_{\text{KL}} + \mathcal{L}_R \quad (4.11)$$

Dla danych ciągłych błąd rekonstrukcji  $\mathcal{L}_R$  odpowiada kwadratowi odległości euklidesowej pomiędzy obrazem wejściowym a obrazem zrekonstruowanym (Ulger, Yuksel i Yilmaz, 2021):

$$\mathcal{L}_R = (x - \tilde{x})^2. \quad (4.12)$$

W implementacjach, gdzie uczenie sieci neuronowej odbywa się na partiach danych w postaci wsadów (ang. *batches*) takich, że ich elementy  $x \in X_s$ ,  $X_s \subset X$ , bardzo często zarówno błąd rekonstrukcji, jak i człon związany z dywergencją Kullbacka-Leiblera wylicza się dla całego wsadu i uśrednia, dzieląc przez jego rozmiar. Dzięki temu błąd rekonstrukcji dla danych, których wartości odpowiadają wielowymiarowemu rozkładowi normalnemu, to po prostu średni błąd kwadratowy (ang. *mean square error*):

$$\text{MSE} = \frac{1}{L} \sum_{i=1}^L (x - \tilde{x})^2, \quad (4.13)$$

gdzie  $L$  – rozmiar partii danych (ang. *batch size*). Natomiast dla danych, których wartości odpowiadają wielowymiarowemu rozkładowi Bernoulliego, błąd rekonstrukcji ma postać binarnej entropii krzyżowej (ang. *binary cross-entropy*) (Kingma i Welling, 2013):

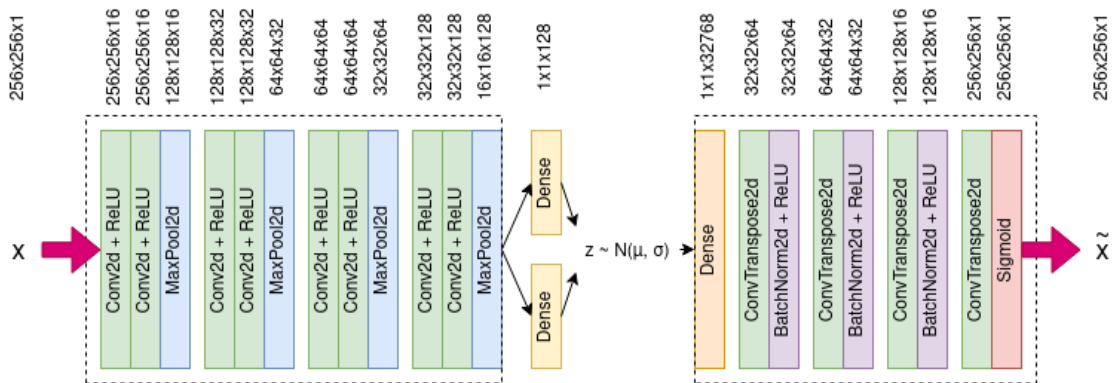
$$\text{BCE} = \sum_{i=1}^L x \log \tilde{x} + (1 - x) \cdot \log(1 - \tilde{x}). \quad (4.14)$$



## 5. Eksperymenty

### 5.1. Model sieci neuronowej i sposób uczenia

Sieć neuronowa stworzona na potrzeby niniejszej pracy ma architekturę typową dla autoenkoderów wariacyjnych. Jej schemat przedstawiono na rysunku 5.1. Całkowita liczba parametrów modelu to 13 010 768.



**Rysunek 5.1.** Architektura autoenkodera wariacyjnego z dekodern Bernoulliego zaimplementowanego na potrzeby niniejszej pracy

Sieć przyjmuje wejście w stałym rozmiarze 256 x 256 pikseli i obsługuje obrazy w skali szarości (tu: zbinaryzowane). Obraz w enkoderze jest przepuszczany przez cztery zestawy warstw w konfiguracji: dwie warstwy splotowe *Conv2d* + warstwa *MaxPool2d* (*max pooling*). W pierwszym zestawie użyto filtrów splotowych o wielkości maski 5 x 5 i wypełnienie (ang. *padding*) równe 2, w kolejnych parametry te zmniejszono do 3 x 3 i 1, odpowiednio. Zastosowano postąpienie (ang. *stride*) równe 1. Pierwszy zestaw warstw zwiększa liczbę kanałów szesnastokrotnie (od 1 do 16), a każdy kolejny już tylko dwukrotnie. Kolejne warstwy *max pooling* zmniejszają rozmiar obrazu o połowę. Tensor wychodzący z autoenkodera, który ma rozmiary 16 x 16 x 128, jest spłaszczany i stanowi wejście do dwóch warstw w pełni połączonych, oznaczonych na ilustracji jako *Dense*. Warstwy te reprezentują średnią i logarytm wariancji. Wykorzystywane są do próbkowania punktu danych ze 128-wymiarowej przestrzeni ukrytej. W dekodern wektor cech jest przekształcany z powrotem do wielowymiarowego tensora stanowiącego mapę cech. W kolejnych warstwach oznaczonych jako *ConvTranspose2d* ma miejsce transponowane splatanie (ang. *transposed convolution*), dzięki czemu powstaje tensor o większych wymiarach. Jest to operacja „wyuczana”, to znaczy, że sieć uczy się, jak rekonstruować obraz o reprezentacji wielowymiarowej z obrazu z niewielką liczbą wymiarów, w przeciwieństwie do zwykłej operacji „upsamplingu”, gdzie przeprowadzana jest z góry ustalana interpolacja. Zastosowana normalizacja wsadowa (ang. *batch normalization*, tu: *BatchNorm2d*) ma stabilizować i zwiększać wydajność uczenia (Ioffe i Szegedy, 2015).

Wyjście z ostatniej warstwy przepuszczane jest przez odpowiednią funkcję aktywacji. Zgodnie z artykułem „*Auto-Encoding Variational Bayes*” (Kingma i Welling, 2013) w dekodern Bernoulliego funkcją tą powinna być funkcja sigmoidalna (przypadek na rys. 5.1).



Dla dekoderek gaussowskich jako funkcję aktywacji po ostatniej warstwie w dekodercie proponuje się użyć tangens hiperboliczny. Co istotne, w zależności od zaimplementowanego rodzaju dekodera błąd rekonstrukcji powinien być wyznaczany w inny sposób, zgodnie z rozważaniami z rozdziału 4.

Sieć uczono z użyciem algorytmu Adam (*Adaptive Moment Estimation*, Kingma i Ba, 2014), rozmiarem wsadu równym 16 elementów, szybkością uczenia  $\alpha = 0,001$ , współczynnikami  $\beta_1$  i  $\beta_2$  równymi odpowiednio 0,5 i 0,999 oraz parametrem  $\epsilon = 10^{-8}$  (oznaczenia zgodne z artykułem autorstwa Kingma i Ba, 2014).

Zbiór danych podzielono w stosunku 85:15 na dwa podzbiory: uczący i walidacyjny, z których to w trakcie uczenia i walidacji losowo wydzielano kolejne wsady (ang. *batches*) w liczbie 16 egzemplarzy. Uczenie przerywano po 200 epokach.

Architektura opracowanego enkodera była zainspirowana architekturami znanych sieci neuronowych, między innymi sieci VGG (Simonyan i Zisserman, 2014). Wybrano jednak większy rozmiar wejścia (256 x 256, a nie 224 x 224), ponieważ jeszcze mniejsze schematy szydełkowe byłyby już nieczytelne dla człowieka. Warstwy dekodera bazowano na przykładowej implementacji z użyciem biblioteki PyTorch (Subramanian, 2020).

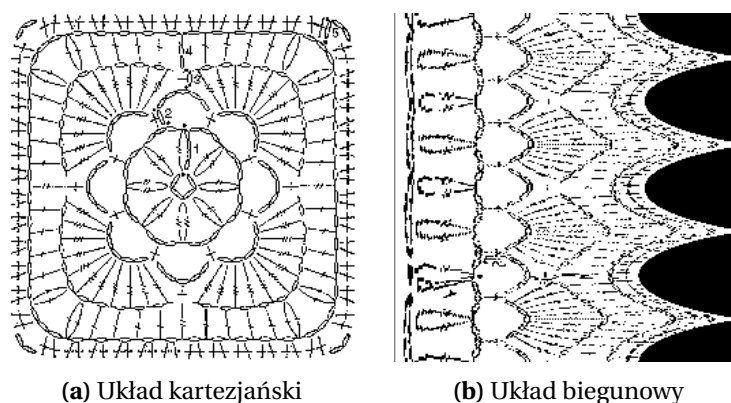
Zgodnie z artykułem "*Auto-Encoding Variational Bayes*" (Kingma i Welling, 2013) w przypadku danych ciągłych należy przyjąć, że  $p_{\theta}(x|z)$  to wielowymiarowy rozkład Gaussa, a w przypadku danych binarnych – wielowymiarowy rozkład Bernoulliego. Dane użyte do obliczeń były zbinaryzowane, ale mimo to przeprowadzono uczenie z dwoma rodzajami dekoderek w celu porównania ich działania.

## 5.2. Augmentacja danych

Ze względu na stosunkowo niewielką liczbę zbioru danych konieczne było jego wzbogacenie. Wybrano metodę augmentacji w trakcie uczenia sieci, tj. losowe zastosowanie wybranych operacji przekształcających obraz w ramach pojedynczej epoki uczącej podczas jego wczytywania z dysku. Wykorzystano zaletę symetrycznych schematów szydełkowych: takie wzory niosą w przybliżeniu (z pominięciem np. pewnych ściegów zamykających) tę samą informację w każdej połówce (i ćwiartce), mają bowiem aż cztery osie symetrii. Dzięki temu możliwe było zastosowanie augmentacji w postaci przerzucenia grafiki w pionie lub poziomie.

Niestety, przekształcenie z kartezjańskiego do biegunowego układu współrzędnych sprawia, że wzory tracą tę właściwość i mają już wtedy tylko jedną oś symetrii, przez co możliwa była tylko augmentacja w pionie. Opisany problem ilustruje rysunek 5.2, na którym widać tę samą grafikę w dwóch różnych układach współrzędnych.

W jednym eksperymencie zastosowano – oprócz wymienionej – specyficzną metodę augmentacji. Z grafik we współrzędnych biegunowych w kierunku pionowym wycięto co pewien krok elementy w postaci kafelków w rozmiarze stanowiącym 1/4 wielkości grafiki wyjściowej. Dzięki temu sztucznie zwiększono liczbę danych uczących, o czym więcej w kolejnych podrozdziałach.



**Rysunek 5.2.** Ten sam schemat szydełkowy w różnych układach współrzędnych

### 5.3. Eksperyment z grafikami w układzie współrzędnych kartezjańskich

Pierwszy eksperyment przeprowadzono na zbiorze danych, którego charakterystykę przedstawiono w rozdziale 3.2. Na potrzeby tego badania grafiki pozostawiono w układzie współrzędnych kartezjańskich. Eksperyment miał potwierdzić hipotezę mówiącą, że nawet za pomocą stosunkowo prostego autoenkodera wariacyjnego z warstwami splotowymi i prostej obróbki danych wejściowych (głównie binaryzacji; jak w rozdziale 3.2) można uzyskać użyteczne wyniki. Weryfikacja tego założenia pozwoliła także na oszacowanie złożoności problemu.

Dzięki tej próbie dopracowano i dostrojono zaproponowany model sieci neuronowej (rys. 5.1), a także ustalono sposób uczenia, który wykorzystano także w późniejszych eksperymentach.

### 5.4. Eksperyment z grafikami w układzie współrzędnych biegunowych

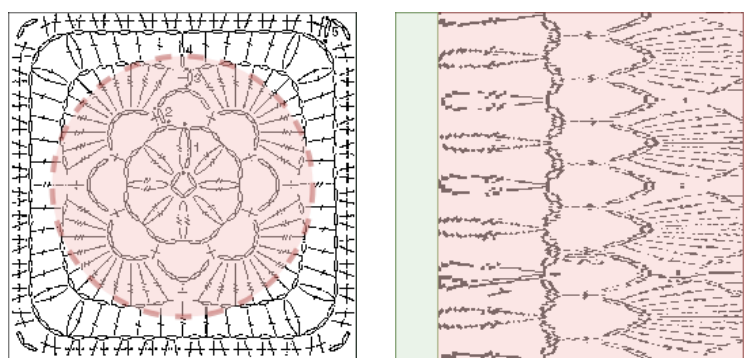
Na potrzeby kolejnego eksperymentu grafiki ze zbioru danych przekształcono z kartezjańskiego do biegunowego układu współrzędnych zgodnie z opisem w rozdziale 4.3. Przykładowy efekt tej operacji widoczny jest na rysunku 5.2. Następnie grafiki podano do tego samego modelu sieci neuronowej, z zachowaniem identycznego sposobu uczenia.

Przekształcenie biegunowe obrazu z użyciem biblioteki OpenCV przebiega jak na rysunku 4.1. Aby objąć przekształceniem cały kwadrat, należy go wpisać w okrąg ograniczający. Wówczas długość promienia okręgu jest oczywiście równa połowie długości przekątnej kwadratu, czyli jest on dłuższy od połowy długości jego boku. Promień „sięga” zatem do obszaru poza obraz, gdzie nie ma już informacji. Ponieważ sieć neuronowa miała ustalony rozmiar wejścia, konieczne było przeprowadzenie transformacji biegunowej, której wynikiem był tensor o tych samych wymiarach co tensor wejściowy. Spowodowało to, że użyta biblioteka OpenCV stłoczyła informację i dopełniła kwadratowy obraz czarnymi obszarami o parabolicznym kształcie. Należy ponadto zaznaczyć, że początek biegunowego układu współrzędnych znajduje się w lewym, górnym rogu obrazu, a sama grafika ma tylko jedną, poziomą oś symetrii. Opisane powody wykluczyły możliwość użycia augmentacji poprzez odbicie grafiki w poziomie, o czym wspomniano wcześniej.

### 5.5. Eksperyment z pominięciem niekorzystnych zjawisk w układzie współrzędnych biegunowych

Rysunek 5.2 pokazuje, że przekształcenie biegunowe prowadzi do rozciągnięcia wewnętrznych rzędów we wzorze szydełkowym, co skutkuje ich uwydatnieniem. Oczka zewnętrzne zostają natomiast na tyle stłoczone, że zawarta tam informacja jest nieczytelna, a przecież instrukcje w środku, jak i na zewnątrz schematu są równie istotne z punktu widzenia procesu szydełkowania.

W kolejnym etapie pracy pominięto opisane niekorzystne zjawiska na krawędziach odpowiadających środkowi układu kartezjańskiego i jego peryferiom. Transformacji biegunowej poddano zatem obszar odpowiadający pewnemu wycinkowi pierścieniowi wokół środka oryginalnej grafiki (rys. 5.3).



(a) Obszar objęty przekształceniem (czerwony) (b) Obszar pominiętej (zielony) i użytecznej informacji (czerwony)

**Rysunek 5.3.** Ilustracja przekształcenia z pominięciem niekorzystnych zjawisk w układzie biegunowym

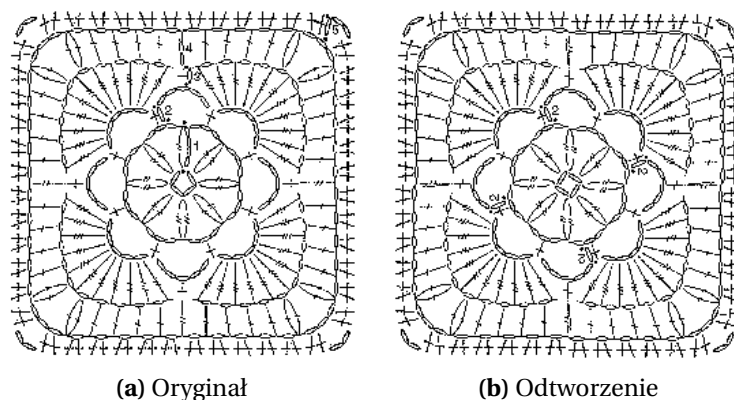
Użyta biblioteka OpenCV pozwoliła na łatwe „odcięcie” zewnętrznych rzędów poprzez sterowanie długością maksymalnego promienia dla transformacji biegunowej. Aby zachować rozmiar obrazu wymuszony przez stały rozmiar wejścia do sieci, wewnętrzne oczka szydełkowe wybielono poprzez „zerowanie” obecnych tam pikseli.

W tym przypadku również zastosowano augmentację jedynie poprzez odbicie w pionie.

### 5.6. Eksperyment z kawałkami grafik w układzie współrzędnych biegunowych

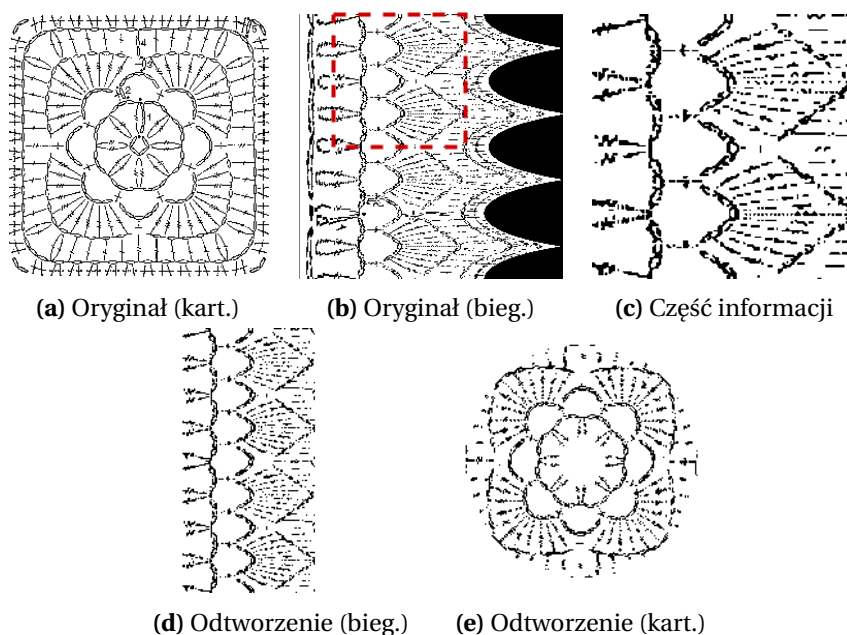
Zbiór danych przygotowany na potrzeby niniejszej pracy składa się ze schematów szydełkowych, które mają cztery osie symetrii. Do odtworzenia całości wzoru z wysoką dokładnością, o ile został on schludnie narysowany przez twórcę, wystarczy dowolna ćwiartka obrazu. Kopiuje się informację obróconą o 90, 180 i 270 stopni i dopełnia obraz. Dokładność jest wówczas wysoka, ale nie stuprocentowa, tym bardziej że pewne oczka zamykające czy rozpoczynające kolejne rzędy znajdują się zazwyczaj tylko w jednej ćwiartce (w jednej z dwóch w górnej połowie obrazu). Przykład opisanego odtworzenia obrazu z jego ćwiartki pokazano na rysunku 5.4.

Grafiki w układzie biegunowym, np. rys. 5.2b, mają tylko jedną oś symetrii. Na podstawie ich ćwiartki nie da się odtworzyć kompletnego schematu. Można natomiast spróbować



**Rysunek 5.4.** Przykład rekonstrukcji wzoru szydełkowego w kartezjańskim układzie współrzędnych z wykorzystaniem jedynie drugiej ćwiartki

wyciąć kwadratowy kafelek w przestrzeni między najbardziej i najmniej skrajnymi rzędami (czyli nie w miejscach, gdzie informacja jest bardzo rozciągnięta lub stłoczona), skopiować go i połączyć dwa takie kawałki pionowo. Przekształcenie uzyskanego obrazu do współrzędnych kartezjańskich pozwoli odtworzyć chociaż część pierwotnej informacji. Ilustruje to rys. 5.5.

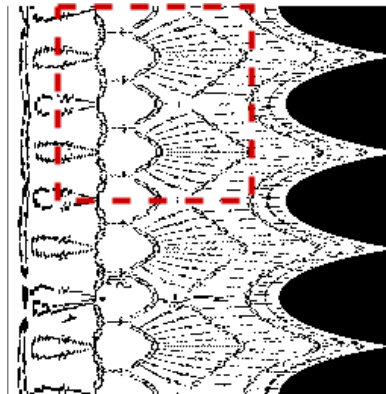


**Rysunek 5.5.** Przykład rekonstrukcji niekompletnej informacji o wzorze szydełkowym z części wyciętej ze schematu w biegunowym układzie współrzędnych

Pomimo tego, że część pikseli, których nie udało się odtworzyć z powodu braku informacji, jest wybielona, otrzymany obrazek wciąż zawiera użyteczną treść.

W świetle powyższego opracowano kolejny eksperyment, podczas którego do modelu sieci neuronowej podawano kwadratowe kafelki wycięte z grafik w układzie biegunowym. Tym razem także pominięto niekorzystne zjawiska w początku układu i na jego peryferiach, odpowiednio skracając maksymalną długość promienia dla transformacji biegunowej i usuwając część informacji w jego początku. Kafelki miały wielkość 1/4 oryginalnego

obrazu (kafelki: 128 x 128 px, obraz pierwotny: 256 x 256 px), dlatego powiększono je do wymiarów 256 x 256 px (konieczność ze względu na rozmiar wejścia do sieci).



**Rysunek 5.6.** Przykład pobierania kwadratowego kafelka ze schematu szydełkowego we współrzędnych biegunowych

Dane wzbogacono (augmentacja) kilkukrotnym wycięciem kafelków z jednego obrazu z niewielkim postąpieniem (2 px) w kierunku poziomym. Oprócz tego zastosowano wcześniej wspomnianą augmentację w pionie.

### 5.7. Eksperymenty z kawałkami grafik w układzie współrzędnych kartezjańskich

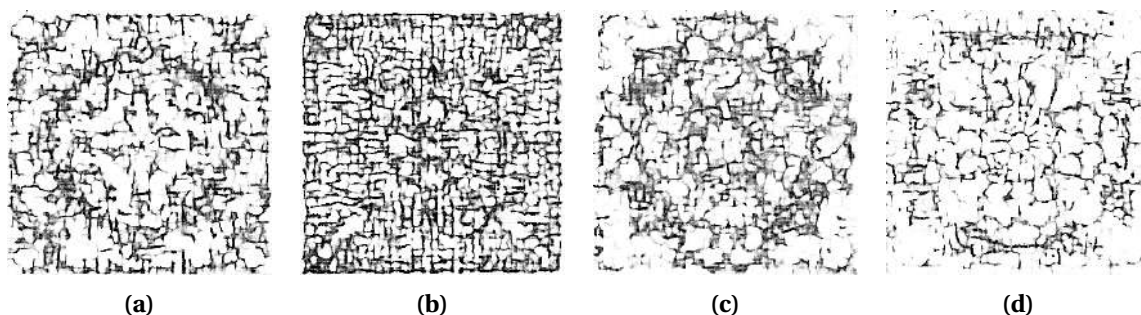
Przeprowadzono także eksperyment z losowo wycinanymi w trakcie uczenia sieci kawałkami grafik w układzie współrzędnych kartezjańskich. Co prawda sztucznie zwiększono liczbę dostępnych danych, ale nie przełożyło się to na wzrost użyteczności wyników. Z powodu ich niskiej wartości nie zamieszczono ich w niniejszej pracy.

## 6. Wyniki

Celem niniejszej pracy było zbadanie możliwości tworzenia wzorów użytkowych za pomocą sieci neuronowych na przykładzie szydełkowania z użyciem obrazkowych wzorów szydełkowych. Aby go zrealizować, przeprowadzono szereg eksperymentów opisanych w rozdziale 5. Podczas każdego z nich sieć próbowała odtworzyć obrazy wejściowe (tj. przygotować rekonstrukcje) i wygenerować zestaw nowych obrazów. Uzyskane wyniki wyostrzono, stosując filtrację. W przypadku eksperymentów w biegunowym układzie współrzędnych obrazy przed wyostrzeniem przekształcono do współrzędnych kartezjańskich.

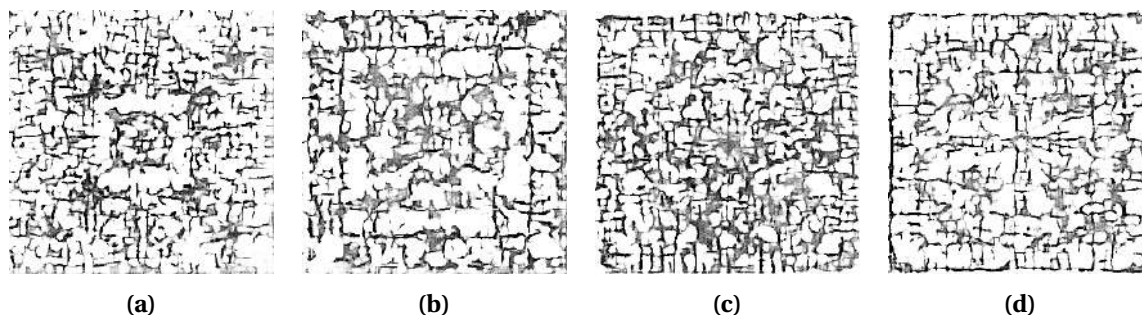
Już najprostszy eksperyment ze zbiorem danych zawierającym obrazy we współrzędnych kartezjańskich przyniósł obiecujące wyniki. Na wygenerowanych obrazach można dostrzec, że sieć poprawnie nauczyła się zasad rządzących schematami szydełkowymi. Widać, że wewnętrzne rzędy są koliste, a im bliżej do brzegu, tym bardziej przypominają kwadrat. Co więcej, sieć poprawnie rozpoznała, że schematy powinny być symetryczne, toteż wyniki mają cztery osie symetrii.

Sieć „dodaje” oczka w rogach, co jest szczególnie widoczne na schematach 6.1a i 6.2b. Można powiedzieć, że poprawnie nauczyła się, że naddania są konieczne, aby zmienić kształt robótki z kolistego na kwadratowy. Niektóre ciemniejsze zbiorowiska pikseli, które zajmują większe obszary, układają się w kształty przypominające pęczki różnego rodzaju słupków (rys. 6.1c). Widoczne połączenie poprzecznym nitkowatymi skupiskami pikseli można interpretować jako ażurowe fragmenty złożone z łańcuszków.



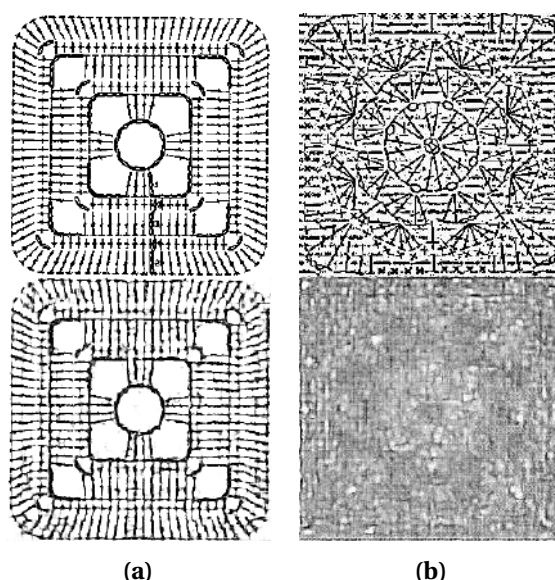
**Rysunek 6.1.** Schematy szydełkowe wygenerowane przez sieć z dekoderym gaussowskim w eksperymencie z grafikami w kartezjańskim układzie współrzędnych

Wygenerowane grafiki nie stanowią jednak kompletnych wzorów szydełkowych. Mimo rysujących się kształtów, które przypominają różne skupiska ściegów, wzory są rozmazane. Rozmazanie jest spowodowane obecnością błędów rekonstrukcji w funkcji straty (wzór 4.9), który bazuje na założeniu, że dostępne dane mają rozkład jednomodalny. Jeżeli jednak jest inaczej, tzn. zbiór danych ma rozkład wielomodalny, autoenkoder nie jest w stanie generować wyrazistych wyników. Problem ten poruszyli w swojej pracy Cai, Gao i Ji (2019), którzy zaproponowali rozwiązanie w postaci „dwustopniowych” dekodery. Zadaniem takich sieci jest najpierw zgrubna generacja, a potem klarowanie obrazu. Innym sposobem wymuszenia wyższej jakości generacji jest np. dołączenie dodatkowych członów regularyzujących do funkcji straty.



**Rysunek 6.2.** Schematy szydełkowe wygenerowane przez sieć z dekodernem Bernoulliego w eksperymencie z grafikami w kartezjańskim układzie współrzędnych

Sieć poprawnie rekonstruuje jedynie proste wzory szydełkowe. Rekonstrukcje najmniej szczegółowych schematów są nieznacznie rozmazane, ale mimo to możliwe jest odczytanie widocznych na nich rodzajów oczek i ściągów (rysunki 6.4a oraz 6.3a). Oczka oznaczone bardzo drobną notacją jak na rys. 6.4b są odtwarzane z trudnością, a wygenerowana grafika zawiera jedynie nieklarowne plamy kształtem przypominające obraz wejściowy.



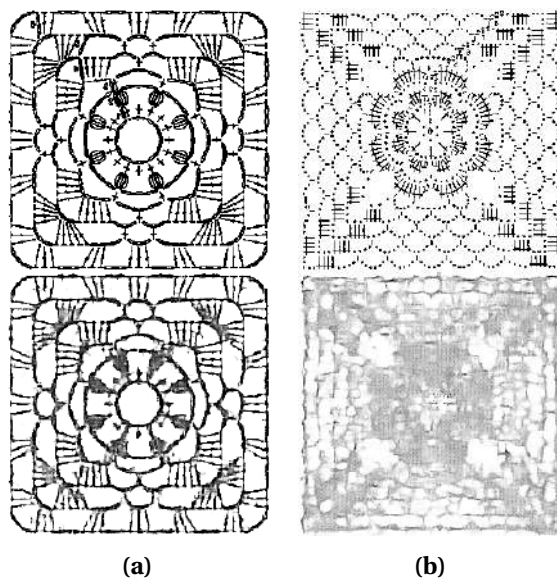
**Rysunek 6.3.** Schematy szydełkowe (góra) i ich rekonstrukcje (dół) przygotowane przez sieć z dekodernem gaussowskim w eksperymencie z grafikami w kartezjańskim układzie współrzędnych

Obecność szumów w danych uczących niemal całkowicie uniemożliwia poprawną rekonstrukcję. Warto zaznaczyć, że zrekonstruowany obraz nigdy nie jest jednak losowy. Mimo dużych zakłóceń widać na nim ciemniejsze kształty przypominające struktury widoczne na wzorze wejściowym.

Jakość wyników uzyskanych z użyciem dekodera Bernoulliego i gaussowskiego jest porównywalna, mimo że intuicyjnie dekodery Bernoulliego powinien być lepszym wyborem dla danych zbinaryzowanych. Ewentualne różnice w wynikach na korzyść któregoś z nich wynikają z losowości.

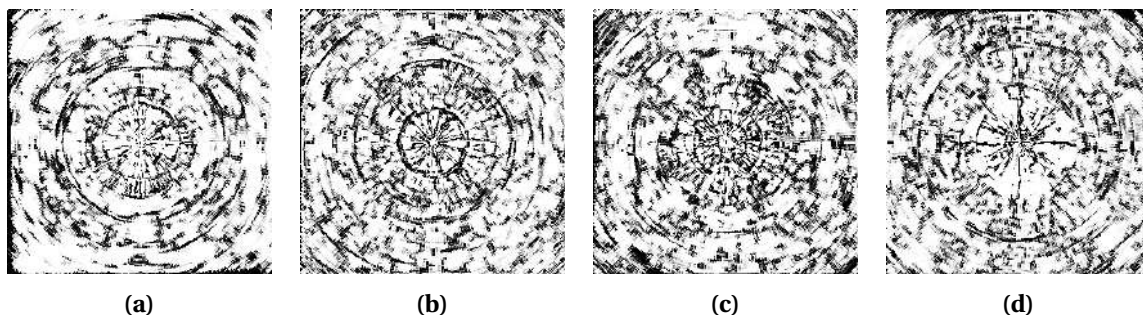
Na obrobionych wynikach eksperymentu z grafikami w biegunowym układzie współrzędnych widać tendencję algorytmu do generowania kolistych struktur. Niemal każdy



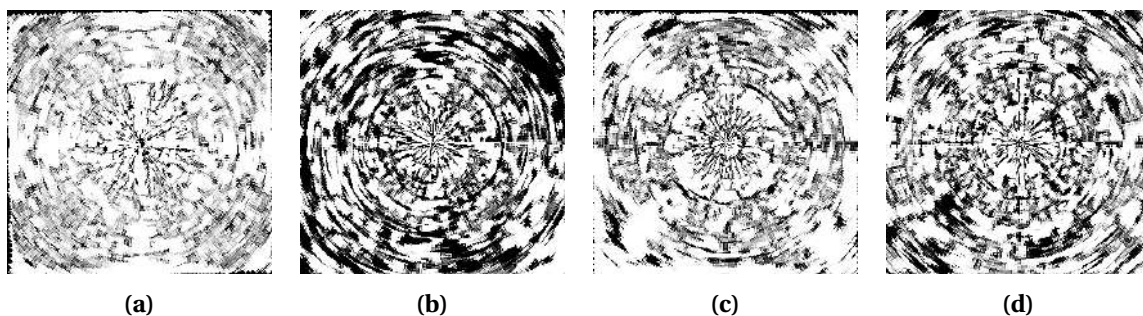


**Rysunek 6.4.** Schematy szydełkowe (góry) i ich rekonstrukcje (dół) przygotowane przez sieć z dekodern Bernoulliego w eksperymencie z grafikami w kartezjańskim układzie współrzędnych

„rząd” ma kształt okręgu, nawet te najbardziej skrajne z nich, mimo że w zbiorze danych dominowały elementy o kwadratowej strukturze.



**Rysunek 6.5.** Schematy szydełkowe wygenerowane przez sieć z dekodernem gaussowskim w eksperymencie z grafikami w biegunowym układzie współrzędnych

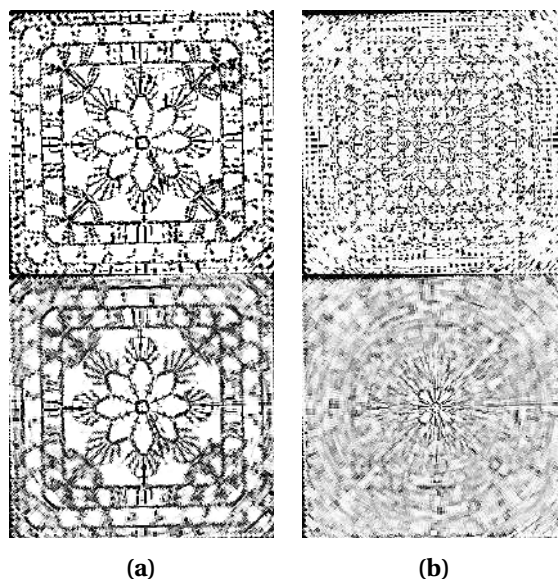


**Rysunek 6.6.** Schematy szydełkowe wygenerowane przez sieć z dekodernem Bernoulliego w eksperymencie z grafikami w biegunowym układzie współrzędnych

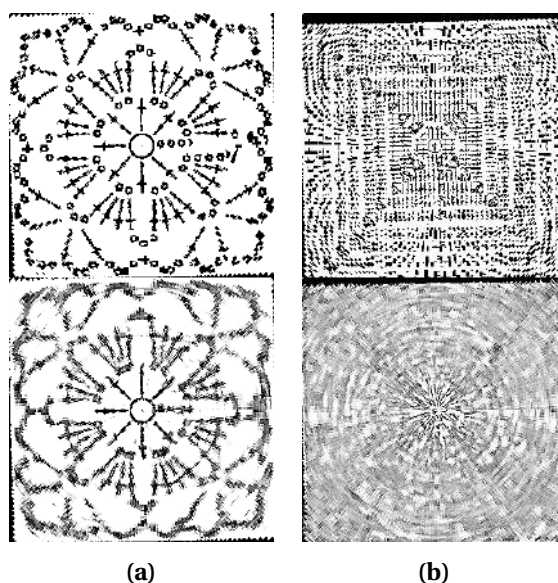
Jedynie środki obrazów są wyraźniejsze, co widać szczególnie w przypadku rekonstrukcji (rys. 6.7, 6.8). Efekty w postaci niewyraźnych plam są widoczne przede wszystkim na brzegach obrazu. Jeżeli schemat szydełkowy przekształca się z kartezjańskiego do biegunowego



układu współrzędnych, umieszczając początek układu w początku schematu, otrzymuje się obraz, na którym wewnętrzne i zewnętrzne rzędy zajmują całą wysokość obrazka w pikselach. Wynika to z faktu, że na oryginalnej „kartezjańskiej” grafice są rozłożone na 360 stopniach. W rezultacie wewnętrzne rzędy stają się rozciągnięte, a zewnętrzne stłoczone, co prowadzi do uwydatnienia części informacji w jednym miejscu i jej utraty w drugim, mimo że jest ona jednakowo ważna wszędzie. Utraconej informacji nie da się odzyskać, co skutkuje pogorszeniem jakości na brzegach grafik po ich przekształceniu z powrotem do układu współrzędnych kartezjańskich.

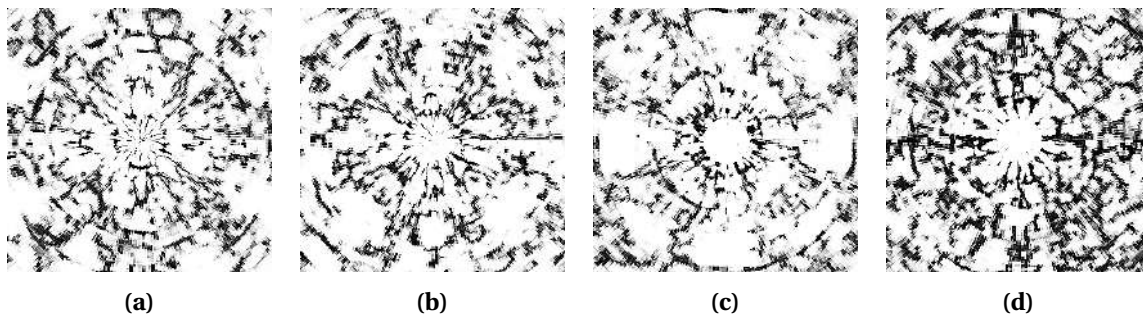


**Rysunek 6.7.** Schematy szydełkowe (góra) i ich rekonstrukcje (dół) przygotowane przez sieć z dekoderym gaussowskim w eksperymencie z grafikami w biegunowym układzie współrzędnych

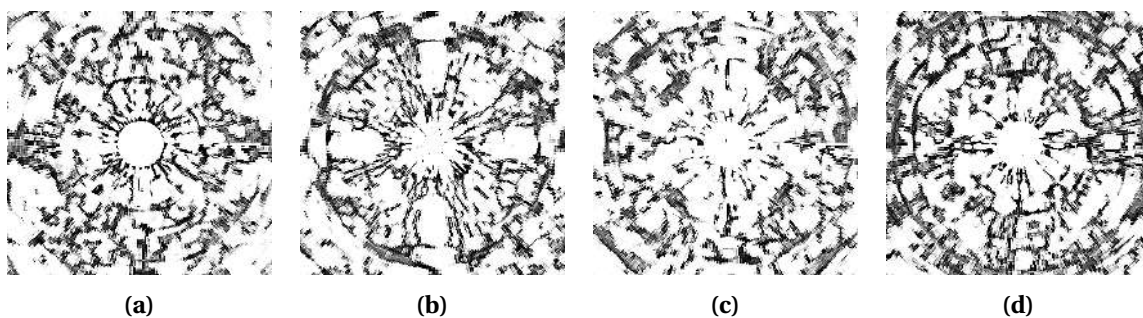


**Rysunek 6.8.** Schematy szydełkowe (góra) i ich rekonstrukcje (dół) przygotowane przez sieć z dekoderym Bernoulliego w eksperymencie z grafikami w biegunowym układzie współrzędnych

Aby zredukować niekorzystne zjawiska na krawędziach odpowiadających środkowi układu kartezjańskiego i jego peryferiom przeprowadzono eksperyment, w którym to sieć uczono na bazie grafik, które zawierały przekształcony do współrzędnych biegunowych obszar odpowiadający pierścieniowi wokół środka oryginalnej grafiki. Skutkiem tego uzyskano schematy, dla których stopień rozmazania jest mniej więcej porównywalny w każdym miejscu.



**Rysunek 6.9.** Wycinki pierścieniowe schematów szydełkowych wygenerowanych przez sieć z dekoderm gaussowskim w eksperymencie z pominięciem niekorzystnych zjawisk w biegunowym układzie współrzędnych

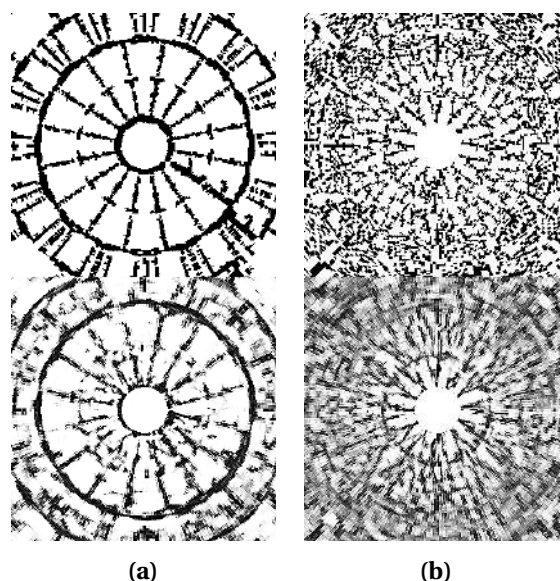


**Rysunek 6.10.** Wycinki pierścieniowe schematów szydełkowych wygenerowanych przez sieć z dekoderm Bernoulliego w eksperymencie z pominięciem niekorzystnych zjawisk w biegunowym układzie współrzędnych

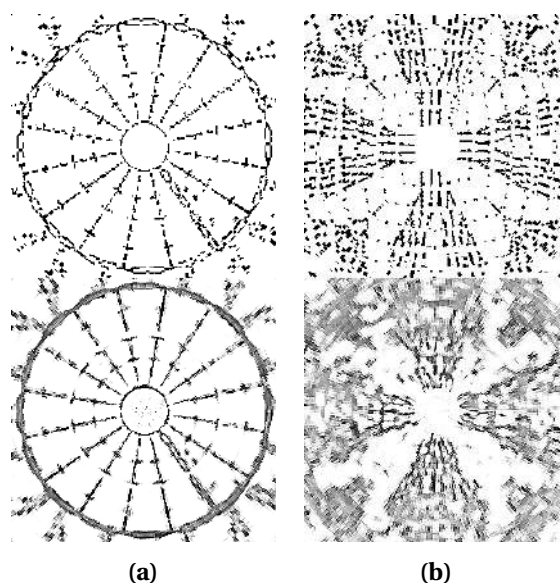
Prawdopodobnie zastosowany zabieg miał także wpływ na to, że sieć wygenerowała bardziej różnorodne wzory, z których wiele ma formę przypominającą kwiat lub krzyż.

Zdolność sieci do generowania bardziej wyszukanych wzorów przełożyła się także na poprawę jakości rekonstrukcji względem poprzedniego eksperymentu. Sieć wprowadziła jednak nieco szumów, przykładowo na rys. 6.9a, w środku obrazów w miejscach, gdzie nie powinna generować informacji. Dane uczące były w tym miejscu wybielone w celu pominięcia „rozciągniętych” informacji w początku wzoru. Dekoder gaussowski częściej wprowadzał szumy niż dekodek Bernoulliego.

Uczenie sieci ze schematami szydełkowymi we współrzędnych kartezjańskich oferuje wyniki o mniej więcej jednorodnym stopniu zaszumienia. Same otrzymane wzory nie są jednak zbyt różnicowane, wręcz są powtarzalne. Sieć wyuczyła się uśrednionej informacji i mimo tego, że w zbiorze zawarte były różne dane, to generacje powstały na podstawie prawdopodobieństwa. Przekształcenie danych uczących do współrzędnych biegunowych,



**Rysunek 6.11.** Wycinki pierścieniowe schematów szydełkowych (górze) i ich rekonstrukcje (dół) przygotowane przez sieć z dekodern gaussowskim w eksperymencie z pominięciem niekorzystnych zjawisk w biegunowym układzie współrzędnych

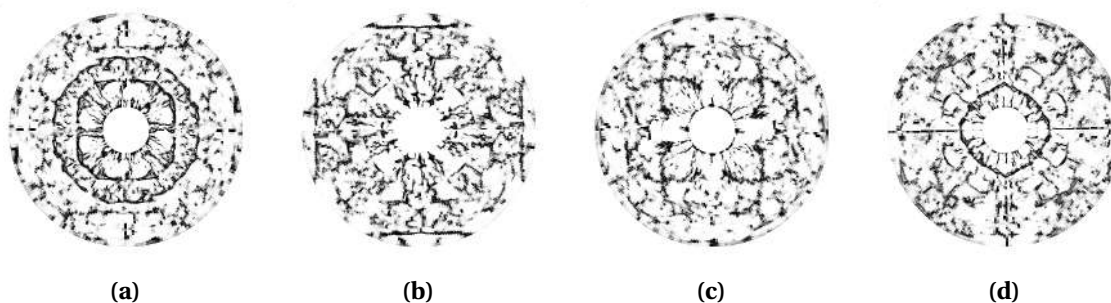


**Rysunek 6.12.** Wycinki pierścieniowe schematów szydełkowych (górze) i ich rekonstrukcje (dół) przygotowane przez sieć z dekodern Bernoulliego w eksperymencie z pominięciem niekorzystnych zjawisk w biegunowym układzie współrzędnych

a następnie podjęcie próby z obszarami odpowiadającymi wycinkom pierścieniowym wokół środka oryginalnej grafiki pomogło wyostrzyć przynajmniej część obrazu. Mimo tego trudno mówić o wzbogaceniu zebranych dotąd wyników o nowe konfiguracje wzorów. Losowa augmentacja w postaci przrzućcia w pionie lub poziomie nie zaradziła małej ilości danych uczących.

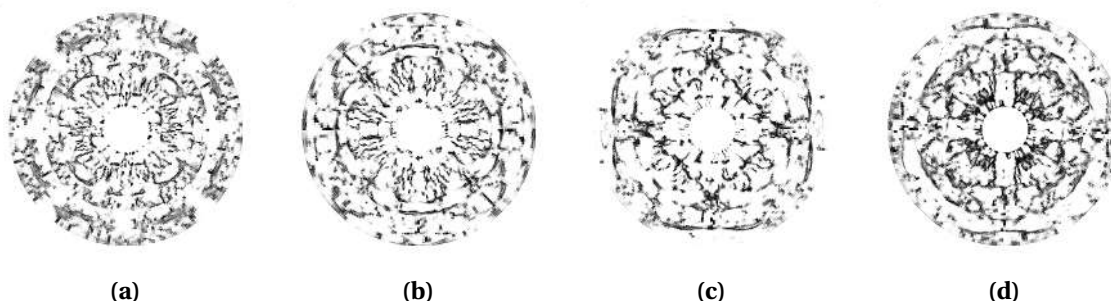
Podniesienie jakości generacji wymagało dodatkowego wzbogacenia zbioru danych uczących. Podawanie do sieci neuronowej kwadratowych kafelków wyciętych z grafik w układzie współrzędnych biegunowych (dokładny opis w rozdziale 5.6) okazało się tutaj

bardzo korzystne. Ponieważ kafelki te były pobierane wielokrotnie z jednej grafiki z niewielkim postąpieniem w kierunku poziomym, na każdym z nich znajdowała się nieco inna porcja informacji dla sieci. Ilość tej informacji była mniej więcej równomiernie rozłożona na małej powierzchni kafelka, ponieważ był on wycinany w pasie pomiędzy najbardziej i najmniej skrajnymi rzędami oryginalnej grafiki. Poskutkowało to osiągnięciem bardziej jednorodnego poziomu ostrości. Otrzymane kafelki przekształcano do układu współrzędnych kartezjańskich zgodnie ze sposobem widocznym na rys. 5.5c-5.5e.



**Rysunek 6.13.** Część informacji o schemacie szydełkowym wygenerowana przez sieć z dekoderm gaussowskim w eksperymencie z kawałkami grafik w układzie biegunowym

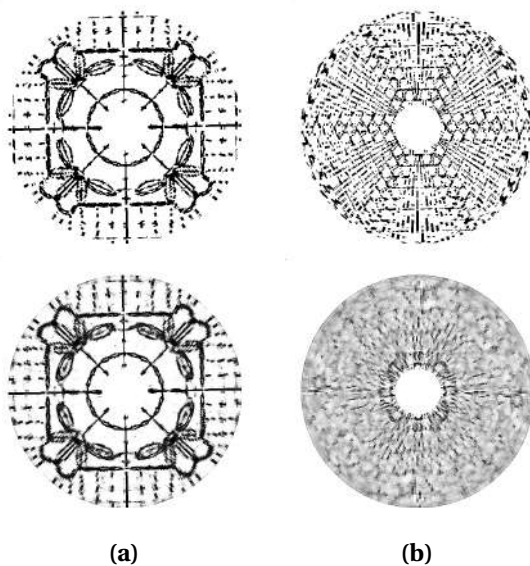
Ocena wygenerowanych schematów podsuwa wniosek, że ostatni eksperyment przyniósł najlepsze wyniki. Otrzymane schematy nie stanowią jedynie prostych kombinacji danych uczących, lecz nowe, różnorodne wzory.



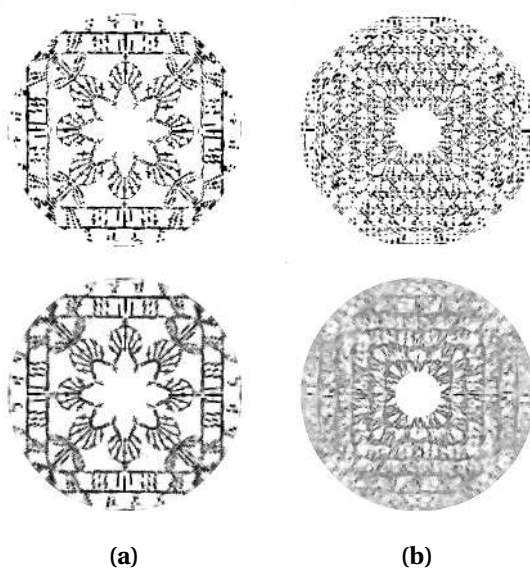
**Rysunek 6.14.** Część informacji o schemacie szydełkowym wygenerowana przez sieć z dekoderm Bernoulliego w eksperymencie z kawałkami grafik w układzie biegunowym

Dzięki dodatkowej formie augmentacji poprawiła się także jakość rekonstrukcji (rys. 6.15, 6.16) – sieć radzi sobie nieco lepiej nawet z trudniejszymi przypadkami. Niestety poziom rekonstrukcji bardzo skomplikowanych wzorów czy też grafik z dużym poziomem szumu wciąż jest niedostateczny.

Rysunki 6.17 i 6.18 przedstawiają dwa wybrane przebiegi uczenia i walidacji sieci z dekoderm Bernoulliego w dwóch różnych eksperymentach. Pierwszy z nich obrazuje wartości funkcji straty dla uczenia z wykorzystaniem grafik we współrzędnych kartezjańskich, a drugi dla uczenia na bazie kafelków z grafik w układzie współrzędnych biegunowych. W obu przypadkach ma miejsce przeuczenie. Jest ono jednak mniej wyraźne w drugim z nich. Zwiększenie zbioru danych zmniejszyło oscylacje wartości funkcji straty w procesie



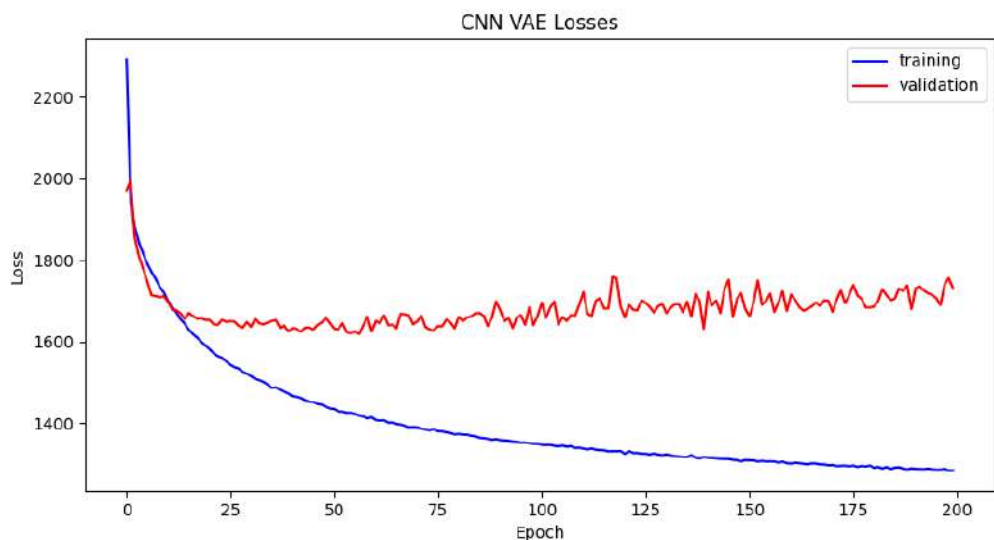
**Rysunek 6.15.** Części informacji o schemacie szydełkowym (górze) i ich rekonstrukcje (dół) przygotowane przez sieć z dekoderym gaussowskim w eksperymencie z kawałkami grafik w układzie biegunowym



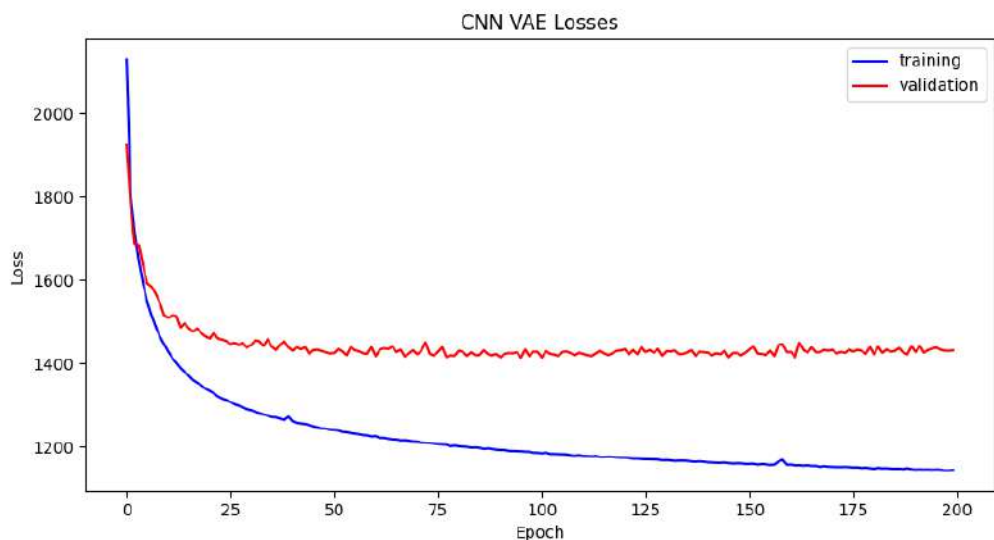
**Rysunek 6.16.** Części informacji o schemacie szydełkowym (górze) i ich rekonstrukcje (dół) przygotowane przez sieć z dekoderym Bernoulliego w eksperymencie z kawałkami grafik w układzie biegunowym

walidacji. Przyczyny przeuczenia sieci należy szukać prawdopodobnie przede wszystkim w ograniczonym rozmiarze zbioru danych.

W niniejszym rozdziale zamieszczono jedynie część generacji i rekonstrukcji uzyskanych w eksperymentach. Kompletnie wyniki dołączono do pracy w formie dodatków.



**Rysunek 6.17.** Przebieg zmian funkcji straty w trakcie uczenia i walidacji sieci z dekodernem Bernoulliego w eksperymencie z grafikami w układzie współrzędnych kartezjańskich



**Rysunek 6.18.** Przebieg zmian funkcji straty w trakcie uczenia i walidacji sieci z dekodernem Bernoulliego w eksperymencie z kawałkami grafik w układzie współrzędnych biegunowych

## 7. Wnioski

Niniejsza praca dotyczyła zbadania możliwości tworzenia wzorów użytkowych za pomocą sieci neuronowych na przykładzie szydełkowania na podstawie schematów obrazkowych. Jest to pierwsze takie badanie w tej dziedzinie, mimo powszechnego zainteresowania wykorzystaniem metod sztucznej inteligencji w sztuce.

Zrealizowanie założonego celu badania o pionierskim charakterze nie byłoby możliwe bez przygotowania odpowiedniego zbioru danych. W związku z tym skompletowano i wstępnie obrobiono schematy szydełkowe znalezione w Internecie. Zebrane obrazy zostały przede wszystkim odpowiednio przycięte ręcznie i zbinaryzowane w celu pozostawienia jedynie informacji istotnej z punktu uczenia sieci neuronowej. Dzięki temu zmniejszono także nakład obliczeniowy.

W tekście przedstawiono także krótką charakterystykę i zasadę działania autoenkoderów wariacyjnych. W kolejnych rozdziałach omówiono stworzony model i sposób jego uczenia. Przeprowadzone badania potwierdzają, że możliwe jest uzyskanie zadowalających wyników z wykorzystaniem prostego modelu sztucznej sieci neuronowej.

Praca ta koncentrowała się głównie wokół metod obróbki i przetwarzania danych wejściowych i wyników w celu poprawy jakości generowanych schematów szydełkowych. Pominięto optymalizację struktur modelu. W przeglądzie literatury zaprezentowano alternatywne podejście do kodowania obrazów obecne w publikacjach medycznych. Bazując na analogiach do kodowania kolistych struktur w układzie współrzędnych biegunowych, opracowano plan eksperymentów poprawiających jakość generacji.

Uczenie sieci neuronowej z użyciem schematów szydełkowych we współrzędnych kartezjańskich produkuje wyniki o mniej więcej jednorodnym poziomie zaszumienia. Przetworzenie zbioru danych do współrzędnych biegunowych nie podnosi jakości generacji. Przekształcenie skutkuje utratą pewnych wiadomości w peryferiach układu kartezjańskiego, a sieć nie radzi sobie z uczeniem się stłoczonych informacji. Ostatecznie na wynikach przekształconych z powrotem do współrzędnych kartezjańskich obecne są koliste struktury stanowiące bardziej szum niż „autentyczną” informację. Opisane niekorzystne zjawiska można zredukować, wprowadzając do sieci neuronowej pewien obszar odpowiadający wycinkowi pierścieniowemu wokół środka oryginalnej grafiki, przekształcony do współrzędnych biegunowych. Poprawia to jakość uzyskanych wyników, ale nie wpływa dodatnio na ich różnorodność. Dopiero dodatkowa augmentacja danych poprzez pobieranie z obrazów we współrzędnych biegunowych wielokrotnych wycinków w postaci kafelków przełożyła się na wzrost złożoności generowanych wzorów.

Autoenkodery wariacyjne mają tendencję do tworzenia niewyraźnych obrazów. Z tej przyczyny uzyskane w ramach badań wyniki nie stanowią kompletnych schematów kwadratów szydełkowych, gdyż nie widać na nich konkretnych oznaczeń oczek i ściągów. Z drugiej strony na wzorach widoczne są skomplikowane kombinacje kształtów, które nie są prostymi złożeniami danych uczących. Informacja układu się w sposób bardziej lub mniej zwarty i zachowuje symetrię, co pozwoli doświadczonemu dziewiarzowi zinterpretować.

wać ją jako konkretne skupiska ściągów. Uzyskane wyniki są zatem źródłem wartościowej inspiracji do tworzenia nowych aranżacji.

Przedstawiona praca poruszyła jedynie część zagadnień i wykorzystała tylko niektóre możliwości oferowane dzisiaj przez sztuczną inteligencję. Badania można byłoby rozwijać z modelami sieci neuronowych o wyższym stopniu zaawansowania i łączyć zalety kodowania obrazów w dwóch układach współrzędnych: kartezyjskim i biegunowym. Przykładowo, w rozdziale 4.3 przedstawiono „dwugałęziową” architekturę sieci neuronowej, która przed dalszym przetwarzaniem danych łączy „kartezyjski” i „biegunowy” wektor cech. Uczenie się komplementarnych informacji o kontekście można byłoby wspomagać specjalnymi modułami, które odpowiadałyby za wymianę i uzgadnianie informacji między „gałęziami” modelu. Literatura dotycząca dziewiarstwa i sztucznej inteligencji sugeruje wykorzystanie sieci z pamięcią. Być może korzystne okazałoby się stworzenie rekurencyjnej sieci neuronowej lub jej połączenie w postaci modułów z warstwami splotowymi. Najbardziej zaawansowane podejście, które intuicyjnie mogłoby oferować wyniki najbliższe pełnowartościowym schematom szydełkowym, to opracowanie modelu sieci neuronowej naśladowującej modele językowe. Sieć taka musiałaby działać w sposób sekwencyjny jak głowica, przesuając się w kierunku odwrotnym do ruchu wskazówek zegara i analizując kolejne oczka, sposoby ich łączenia oraz reguły rządzące schematami. Dopiero na tej podstawie powinna uczyć się, jak przewidywać oczka konieczne do wydziergania w kolejnym kawałku robótki. Być może odzwierciedlenie sposobu szydełkowania w zasadzie działania sieci byłoby właściwym podejściem. Aby uzyskać schematy, na których widoczne byłyby konkretne oczka i ich konfiguracje, należałoby w pełni wykorzystać wiedzę dziedzinową i stworzyć „słownik”, który pozwoliłby uzgodnić notację pomiędzy różnymi schematami.

W świetle powyższego widać, że istnieje jeszcze wiele możliwości poprawy generacji wzorów szydełkowych. Ten rodzaj dziewiarstwa jest jednak niemożliwy do zautomatyzowania i podrobienia przez maszynę, w przeciwieństwie do robienia na drutach. W związku z tym nawet niekompletne schematy szydełkowe są użyteczne, ponieważ mogą stanowić istotne źródło inspiracji dla człowieka.



## Bibliografia

- 10 *crochet granny square* (2021). W: [turquoisewithvanilla.com](https://turquoisewithvanilla.com) [online]. Dostępny w Internecie: <https://turquoisewithvanilla.com/2018/10/29/10-crochet-granny-square-free-patterns/>. Dostęp: 11.08.2023.
- Benčević, Marin i in. (2021). „Training on polar image transformations improves biomedical image segmentation”. W: *IEEE Access* 9, s. 133365–133375.
- Biesiada, Marek (2014). *Statystyka w ujęciu Bayesowskim*. W: Skrypty dla studentów Ekonofizyki na Uniwersytecie Śląskim, Uniwersytet Śląski w Katowicach. Dostępny w Internecie: [https://el.us.edu.pl/ekonofizyka/index.php/Statystyka\\_w\\_uj%C4%99ciu\\_Bayesowskim](https://el.us.edu.pl/ekonofizyka/index.php/Statystyka_w_uj%C4%99ciu_Bayesowskim). Dostęp: 21.08.2023.
- Boho granny square pattern* (2019). W: [bohemian-cat.com](http://bohemian-cat.com) [online]. Dostępny w Internecie: <http://bohemian-cat.com/granny-square-pattern-for-beginners/>. Dostęp: 10.08.2023.
- Cai, Lei, Hongyang Gao i Shuiwang Ji (2019). „Multi-stage variational auto-encoders for coarse-to-fine image generation”. W: *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM, s. 630–638.
- Cheng, Xuemin i in. (2020). „Method for training convolutional neural networks for in situ plankton image recognition and classification based on the mechanisms of the human eye”. W: *Sensors* 20.9, s. 2592.
- Crochet chain stitch* (2022). W: [Amigurumi.com](https://www.amigurumi.com) [online]. Dostępny w Internecie: <https://www.amigurumi.com/stitches/chain/>. Dostęp: 10.08.2023.
- Fabius, Otto i Joost R Van Amersfoort (2014). „Variational recurrent auto-encoders”. W: *arXiv preprint arXiv:1412.6581*.
- Feng, Yaowei i in. (2022). „Polarformer: Optic Disc and Cup Segmentation Using a Hybrid CNN-Transformer and Polar Transformation”. W: *Applied Sciences* 13.1, s. 541.
- Goodfellow, Ian, Yoshua Bengio i Aaron Courville (2016). „Chapter 20: Deep Generative Models”. W: *Deep Learning; MIT Press: Cambridge, MA, USA*, s. 654–720.
- Gregor, Karol i in. (2015). „Draw: A recurrent neural network for image generation”. W: *International conference on machine learning*. PMLR, s. 1462–1471.
- Ioffe, Sergey i Christian Szegedy (2015). „Batch normalization: Accelerating deep network training by reducing internal covariate shift”. W: *International conference on machine learning*. pmlr, s. 448–456.
- Jak szydełkować po amerykańsku* (2015). W: [Magic Carpet Studio](http://magiccarpetstudio.blogspot.com) [online]. Dostępny w Internecie: <http://magiccarpetstudio.blogspot.com/2015/02/jak-szydekowac-po-amerykansku.html>. Dostęp: 14.08.2023.
- Kaspar, Alexandre, Liane Makatura i Wojciech Matusik (2019). „Knitting skeletons: A computer-aided design tool for shaping and patterning of knitted garments”. W: *proceedings of the 32nd annual ACM symposium on user interface software and technology*, s. 53–65.
- Kaspar, Alexandre, Tae-Hyun Oh i in. (2019). „Neural inverse knitting: From images to manufacturing instructions”. W: *International Conference on Machine Learning*. PMLR, s. 3272–3281.

- Kim, B-S i in. (2018). „CNN-based UGS method using Cartesian-to-polar coordinate transformation”. W: *Electronics Letters* 54.23, s. 1321–1322.
- Kim, Jinpyo i in. (2020). „CyCNN: A rotation invariant CNN using polar mapping and cylindrical convolution layers”. W: *arXiv preprint arXiv:2007.10588*.
- Kingma, Diederik P i Jimmy Ba (2014). „Adam: A method for stochastic optimization”. W: *arXiv preprint arXiv:1412.6980*.
- Kingma, Diederik P i Max Welling (2013). „Auto-encoding variational bayes”. W: *arXiv preprint arXiv:1312.6114*.
- Kingma, Diederik P, Max Welling i in. (2019). „An introduction to variational autoencoders”. W: *Foundations and Trends® in Machine Learning* 12.4, s. 307–392.
- Lesson 10: How to Crochet Turning Chain* (brak daty). W: Annie's Craft Store [online]. Dostępny w Internecie: [https://www.anniescatalog.com/crochet/content.html?content\\_id=595&type\\_id=S](https://www.anniescatalog.com/crochet/content.html?content_id=595&type_id=S). Dostęp: 10.08.2023.
- Liu, Qing i in. (2019). „DDNet: cartesian-polar dual-domain network for the joint optic disc and cup segmentation”. W: *arXiv preprint arXiv:1904.08773*.
- Madrigal, Alexis C. (2018). *SkyKnit: How an AI Took Over an Adult Knitting Community*. W: The Atlantic [online]. Dostępny w Internecie: <https://www.theatlantic.com/technology/archive/2018/03/the-making-of-skyknit-an-ai-yarn/554894/>. Dostęp: 14.08.2023.
- Magic ring crochet* (2022). W: Amigurumi.com [online]. Dostępny w Internecie: <https://www.amigurumi.com/stitches/magic-ring/>. Dostęp: 10.08.2023.
- Odaibo, Stephen (2019). „Tutorial: Deriving the standard variational autoencoder (vae) loss function”. W: *arXiv preprint arXiv:1907.08956*.
- OpenCV (2015). *Geometric Image Transformations*. Dostępny w Internecie: [https://docs.opencv.org/3.4/da/d54/group\\_\\_imgproc\\_\\_transform.html](https://docs.opencv.org/3.4/da/d54/group__imgproc__transform.html). Dostęp: 23.08.2023.
- Ryj (2008). *Maschenbildung mit Spitzennadeln*. W: Wikimedia Commons [online]. Dostępny w Internecie: [https://commons.wikimedia.org/wiki/File:Maschenbildung\\_1.jpg](https://commons.wikimedia.org/wiki/File:Maschenbildung_1.jpg). Dostęp: 11.08.2023.
- Shane, Janelle (2019). *First there was SkyKnit. Now there's HAT3000*. W: AI Weirdness [online]. Dostępny w Internecie: <https://www.aiweirdness.com/first-there-was-skyknit-now-theres-19-09-04/>. Dostęp: 14.08.2023.
- Simonyan, Karen i Andrew Zisserman (2014). „Very deep convolutional networks for large-scale image recognition”. W: *arXiv preprint arXiv:1409.1556*.
- Subramanian, A.K (2020). *PyTorch-VAE*. Dostępny w Internecie: <https://github.com/AntixK/PyTorch-VAE>. Dostęp: 14.08.2023.
- Ulger, Furkan, Seniha Esen Yuksel i Atila Yilmaz (2021). „Anomaly detection for solder joints using  $\beta$ -VAE”. W: *IEEE Transactions on Components, Packaging and Manufacturing Technology* 11.12, s. 2214–2221.
- US Crochet Stitch Terms* (2018). W: <http://blog.illuminecreative.com> [online]. Dostępny w Internecie: <http://blog.illuminecreative.com/about/>. Dostęp: 11.08.2023.

# Wykaz symboli i skrótów

## Symbole matematyczne

$D_{KL}$  – dywergencja Kullbacka-Leiblera

$E_{\sim q_{\phi}(z|x_i)}$  – oczekiwany błąd rekonstrukcji

$H$  – wysokość obrazu

$\mathcal{L}$  – funkcja celu autoenkodera wariacyjnego;  $\mathcal{L}(\phi, \theta)$  – jako funkcja parametrów  $\phi$  i  $\theta$

$\mathcal{L}_{KL}$  – człon w równaniu funkcji straty związany z dywergencją Kullbacka-Leiblera

$\mathcal{L}_R$  – człon w równaniu funkcji straty związany z błędem rekonstrukcji

$N(0, I)$  – rozkład normalny standardowy

$N(\mu, \sigma)$  – rozkład normalny z wartością oczekiwaną  $\mu$  i odchyleniem standardowym  $\sigma$

$p_{\theta}(x)$  – prawdopodobieństwo obserwowanych danych

$p_{\theta}(z)$  – prawdopodobieństwo a priori

$p_{\theta}(x|z)$  – wiarygodność

$q_{\phi}(z|x)$  – przybliżony rozkład a posteriori

$R$  – maksymalna długość promienia okręgu ograniczającego w przekształceniu biegunowym

$W$  – szerokość obrazu

$x$  – odcięta punktu na obrazie lub dana wejściowa

$x_0$  – odcięta środka obrazu w układzie współrzędnych kartezjańskich

$\tilde{x}$  – obraz zrekonstruowany przez dekodera

$X$  – zbiór danych

$X_s$  – podzbiór zbioru danych, pojedynczy wsad (*batch*)

$y$  – rzędna punktu na obrazie

$y_0$  – rzędna środka obrazu w układzie współrzędnych kartezjańskich

$z$  – zmienna ukryta

$\epsilon$  – losowa zmienna próbkowana z rozkładu normalnego standardowego na potrzeby reparametryzacji

$\theta$  – kąt w biegunowym układzie współrzędnych lub wagi dekodera

$\theta^*$  – optymalne wagi dekodera

$\mu$  – średnia

$\rho$  – promień w biegunowym układzie współrzędnych

$\sigma$  – odchylenie standardowe

$\phi$  – wagi enkodera

$\phi^*$  – optymalne wagi enkodera

## Skróty

BCE – *binary cross-entropy*, binarna entropia krzyżowa

ELBO – *evidence lower bound, variational lower bound*, dolne ograniczenie wariacyjne

MSE – *mean squared error*, średni błąd kwadratowy

## Spis rysunków

3.1	Sposób wykonywania kolorowego kwadratu szydełkowego z kwiatowym wizerunkiem ( <i>Boho granny square pattern</i> , 2019) . . . . .	8
3.2	Technika pracy szydełkiem – łańcuszek ( <i>Crochet chain stitch</i> , 2022) . . . . .	9
3.3	Technika pracy szydełkiem – pętelka zaciskowa ( <i>Magic ring crochet</i> , 2022) . . .	10
3.4	Różne ścięgi szydełkowe. Od prawej: oczko ścisłe (nieoznaczone), półsłupek (1), półsłupek nawijany (2), słupek pojedynczy (3), słupek podwójny (4) ( <i>Lesson 10: How to Crochet Turning Chain</i> , brak daty) . . . . .	10
3.5	Cykl pracy maszyny dziewiarskiej (Ryj, 2008) . . . . .	11
3.6	Instrukcje do wykonania prostego kwadratu szydełkowego ( <i>10 crochet granny square</i> , 2021) . . . . .	12
3.7	Sposób wykonywania słupka pojedynczego ( <i>US Crochet Stitch Terms</i> , 2018) . .	12
3.8	Przykłady schematów z różną notacją słupków i półsłupków . . . . .	14
3.9	Przykładowe sposoby przedstawienia początku schematu szydełkowego w postaci łańcuszka . . . . .	15
3.10	Przykład wzoru symetrycznego (z czterema osiami symetrii) i niesymetrycznego	16
3.11	Przykłady zbinaryzowanych wzorów ze zbioru danych . . . . .	16
4.1	Przekształcenie biegunowe w bibliotece OpenCV . . . . .	19
4.2	Zależność pomiędzy kartezjańskim i biegunowym układem współrzędnych .	20
4.3	Przykład dwugąłzowej architektury sieci neuronowej do zadania klasyfikacji (Cheng i in., 2020) . . . . .	21
5.1	Architektura autoenkodera wariacyjnego z dekodерem Bernoulliego zaimplementowanego na potrzeby niniejszej pracy . . . . .	24
5.2	Ten sam schemat szydełkowy w różnych układach współrzędnych . . . . .	26
5.3	Ilustracja przekształcenia z pominięciem niekorzystnych zjawisk w układzie biegunowym . . . . .	27
5.4	Przykład rekonstrukcji wzoru szydełkowego w kartezjańskim układzie współrzędnych z wykorzystaniem jedynie drugiej ćwiartki . . . . .	28
5.5	Przykład rekonstrukcji niekompletnej informacji o wzorze szydełkowym z części wyciętej ze schematu w biegunowym układzie współrzędnych . . . . .	28
5.6	Przykład pobierania kwadratowego kafelka ze schematu szydełkowego we współrzędnych biegunowych . . . . .	29
6.1	Schematy szydełkowe wygenerowane przez sieć z dekodерem gaussowskim w eksperymencie z grafikami w kartezjańskim układzie współrzędnych . . . . .	30
6.2	Schematy szydełkowe wygenerowane przez sieć z dekodерem Bernoulliego w eksperymencie z grafikami w kartezjańskim układzie współrzędnych . . . . .	31
6.3	Schematy szydełkowe (góра) i ich rekonstrukcje (dół) przygotowane przez sieć z dekodерem gaussowskim w eksperymencie z grafikami w kartezjańskim układzie współrzędnych . . . . .	31

6.4	Schematy szydełkowe (góra) i ich rekonstrukcje (dół) przygotowane przez sieć z dekodere Bernoulliego w eksperymencie z grafikami w kartezyjskim układzie współrzędnych . . . . .	32
6.5	Schematy szydełkowe wygenerowane przez sieć z dekodere gaussowskim w eksperymencie z grafikami w biegunowym układzie współrzędnych . . . . .	32
6.6	Schematy szydełkowe wygenerowane przez sieć z dekodere Bernoulliego w eksperymencie z grafikami w biegunowym układzie współrzędnych . . . . .	32
6.7	Schematy szydełkowe (góra) i ich rekonstrukcje (dół) przygotowane przez sieć z dekodere gaussowskim w eksperymencie z grafikami w biegunowym układzie współrzędnych . . . . .	33
6.8	Schematy szydełkowe (góra) i ich rekonstrukcje (dół) przygotowane przez sieć z dekodere Bernoulliego w eksperymencie z grafikami w biegunowym układzie współrzędnych . . . . .	33
6.9	Wycinki pierścieniowe schematów szydełkowych wygenerowanych przez sieć z dekodere gaussowskim w eksperymencie z pominięciem niekorzystnych zjawisk w biegunowym układzie współrzędnych . . . . .	34
6.10	Wycinki pierścieniowe schematów szydełkowych wygenerowanych przez sieć z dekodere Bernoulliego w eksperymencie z pominięciem niekorzystnych zjawisk w biegunowym układzie współrzędnych . . . . .	34
6.11	Wycinki pierścieniowe schematów szydełkowych (góra) i ich rekonstrukcje (dół) przygotowane przez sieć z dekodere gaussowskim w eksperymencie z pominięciem niekorzystnych zjawisk w biegunowym układzie współrzędnych	35
6.12	Wycinki pierścieniowe schematów szydełkowych (góra) i ich rekonstrukcje (dół) przygotowane przez sieć z dekodere Bernoulliego w eksperymencie z pominięciem niekorzystnych zjawisk w biegunowym układzie współrzędnych	35
6.13	Część informacji o schemacie szydełkowym wygenerowana przez sieć z dekodere gaussowskim w eksperymencie z kawałkami grafik w układzie biegunowym . . . . .	36
6.14	Część informacji o schemacie szydełkowym wygenerowana przez sieć z dekodere Bernoulliego w eksperymencie z kawałkami grafik w układzie biegunowym . . . . .	36
6.15	Części informacji o schemacie szydełkowym (góra) i ich rekonstrukcje (dół) przygotowane przez sieć z dekodere gaussowskim w eksperymencie z kawałkami grafik w układzie biegunowym . . . . .	37
6.16	Części informacji o schemacie szydełkowym (góra) i ich rekonstrukcje (dół) przygotowane przez sieć z dekodere Bernoulliego w eksperymencie z kawałkami grafik w układzie biegunowym . . . . .	37
6.17	Przebieg zmian funkcji straty w trakcie uczenia i walidacji sieci z dekodere Bernoulliego w eksperymencie z grafikami w układzie współrzędnych kartezyjskich . . . . .	38

6.18 Przebieg zmian funkcji straty w trakcie uczenia i walidacji sieci z dekodern Bernoulliego w eksperymencie z kawałkami grafik w układzie współrzędnych biegunowych . . . . .	38
--	----

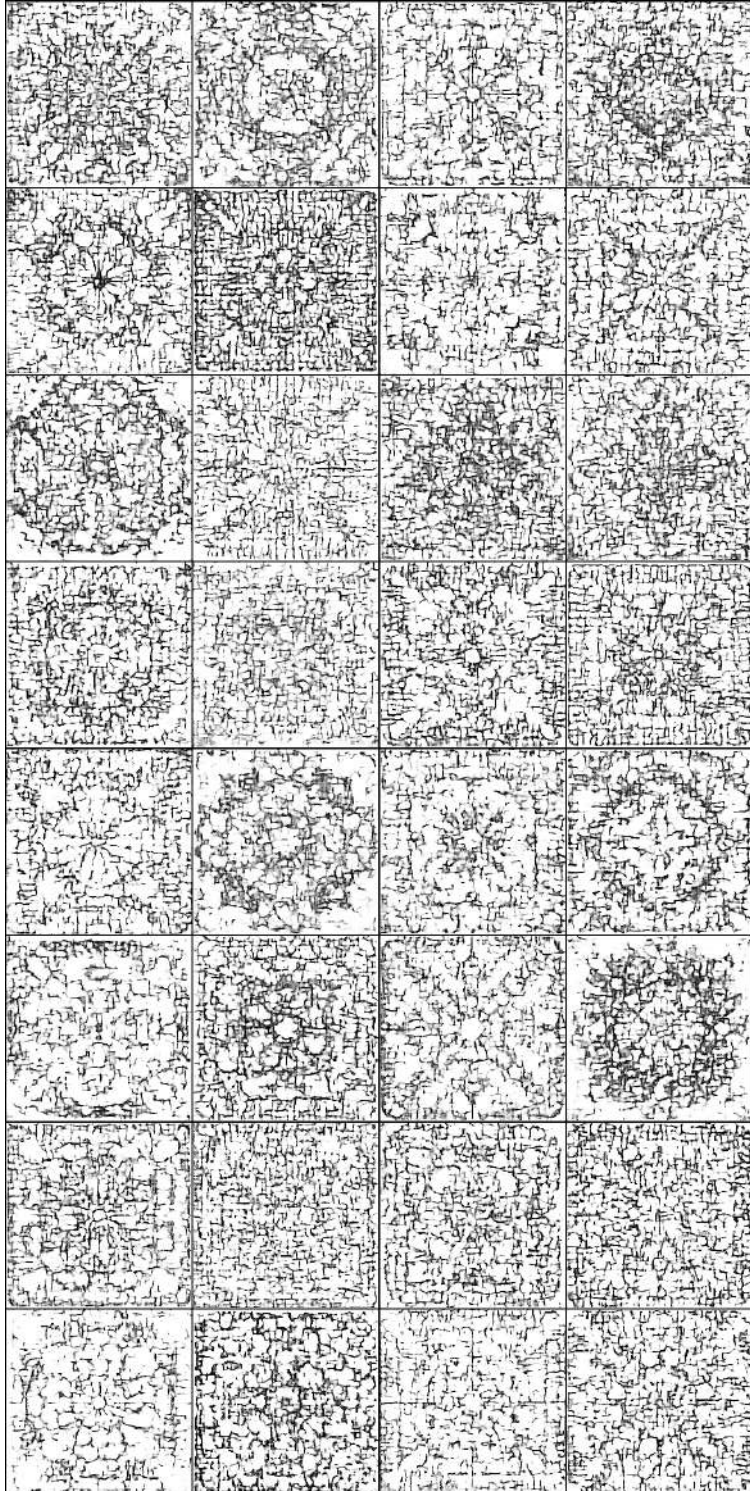
## Spis tabel

3.1 Porównanie nazewnictwa podstawowych ściegów szydełkowych w języku polskim oraz w amerykańskiej i brytyjskiej odmianie języka angielskiego (wraz ze skrótami) ( <i>Jak szydełkować po amerykańsku</i> , 2015) . . . . .	13
3.2 Najczęściej spotykane oznaczenia graficzne podstawowych ściegów szydełkowych . . . . .	14

## Spis dodatków

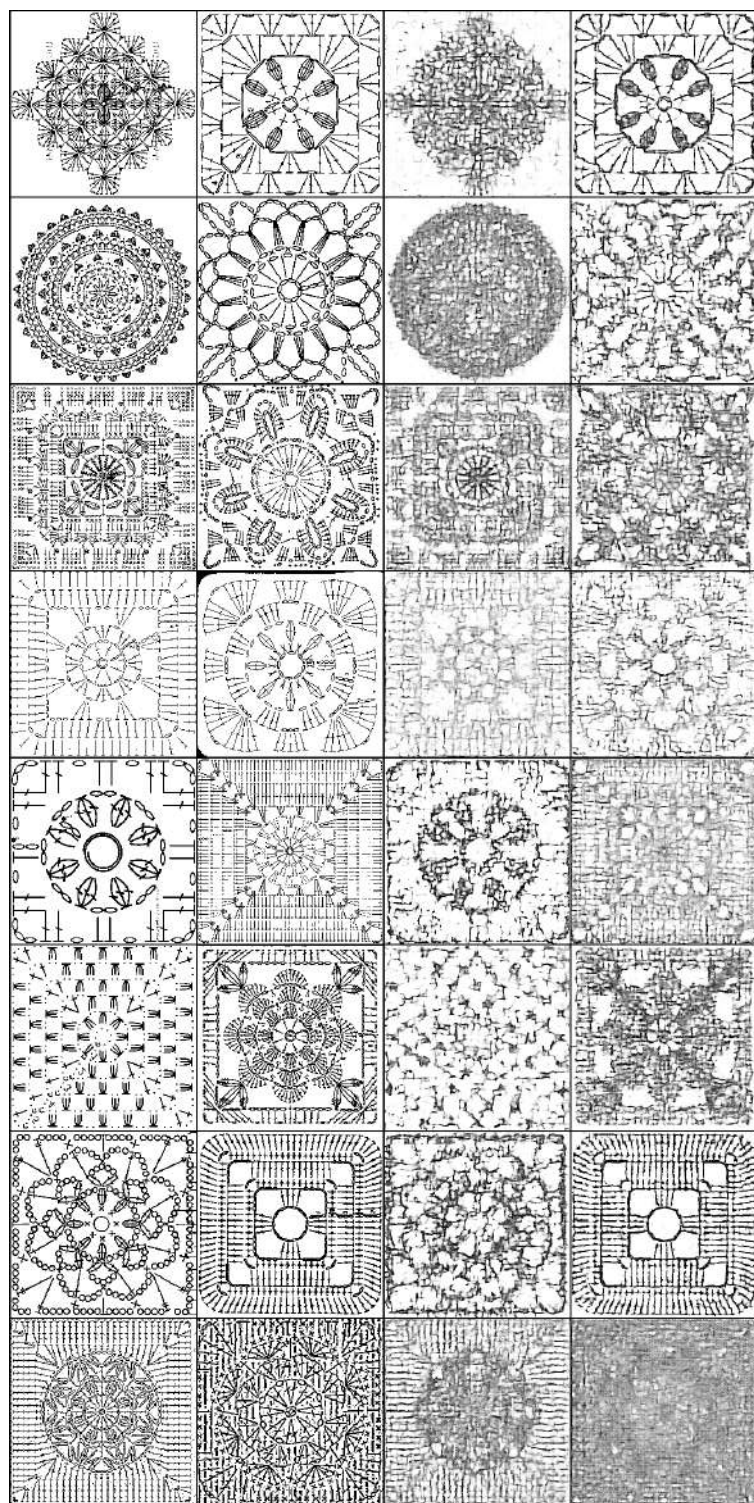
1. Wyniki eksperymentu z grafikami w układzie współrzędnych kartezjańskich . .	47
2. Wyniki eksperymentu z grafikami w układzie współrzędnych biegunowych . . .	52
3. Eksperyment z pominięciem niekorzystnych zjawisk w układzie współrzędnych biegunowych . . . . .	57
4. Wyniki eksperymentu z kawałkami grafik w układzie współrzędnych biegunowych	62

## **Dodatek 1. Wyniki eksperymentu z grafikami w układzie współrzędnych kartezjańskich**

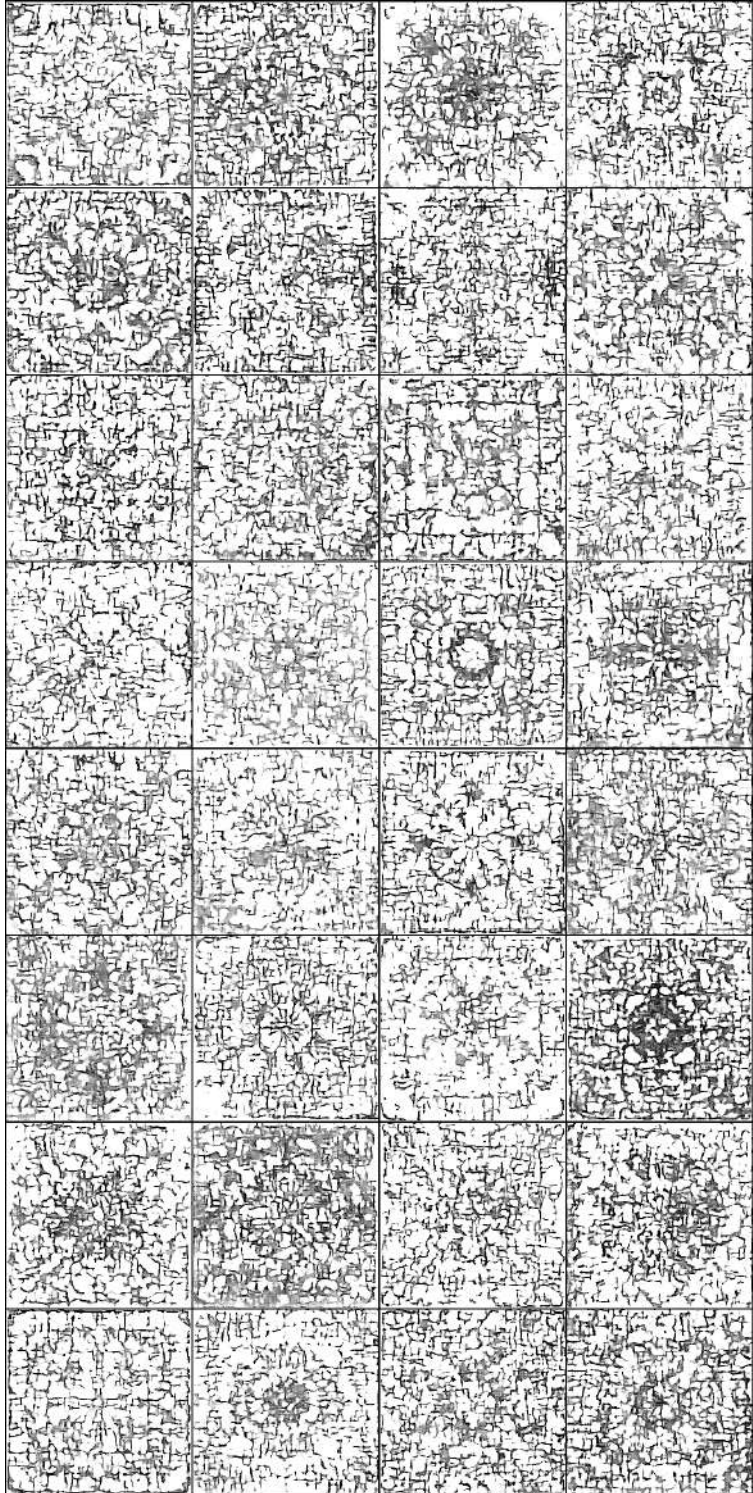


**Rysunek 1.1.** Schematy szydełkowe wygenerowane przez sieć z dekodertem gaussowskim

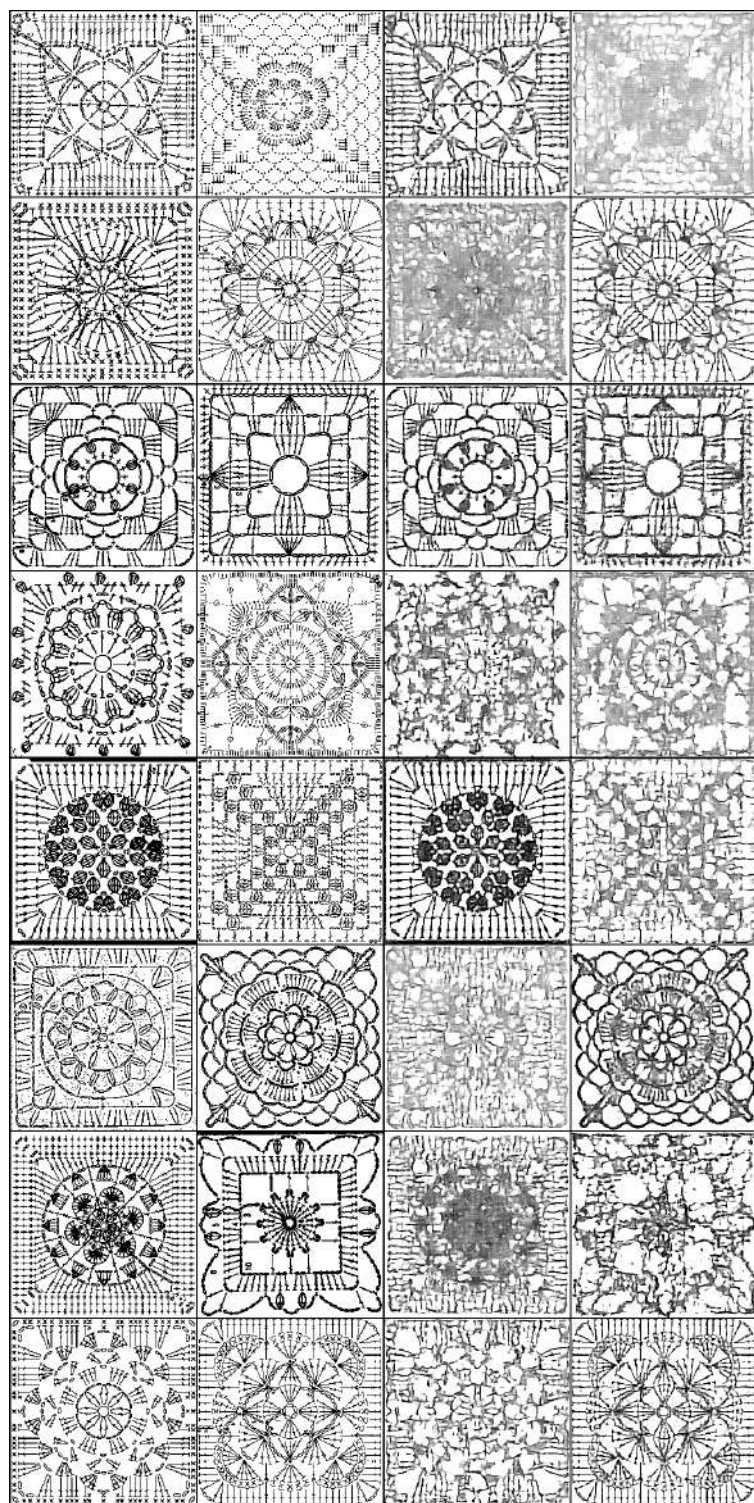




**Rysunek 1.2.** Schematy szydełkowe (dwa górne rzędy) i ich rekonstrukcje (dwa dolne rzędy) przygotowane przez sieć z dekodelem gaussowskim

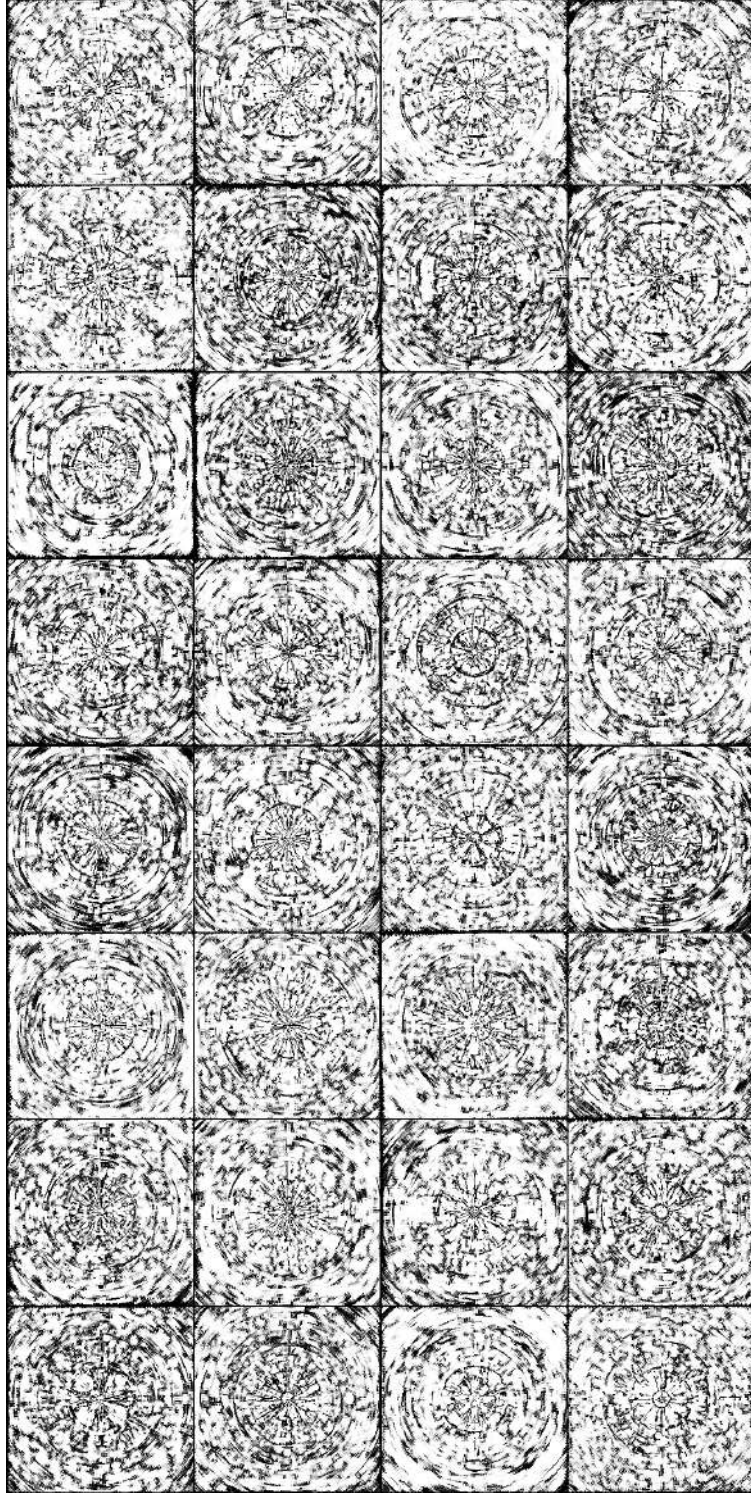


**Rysunek 1.3.** Schematy szydełkowe wygenerowane przez sieć z dekodernem Bernoulliego



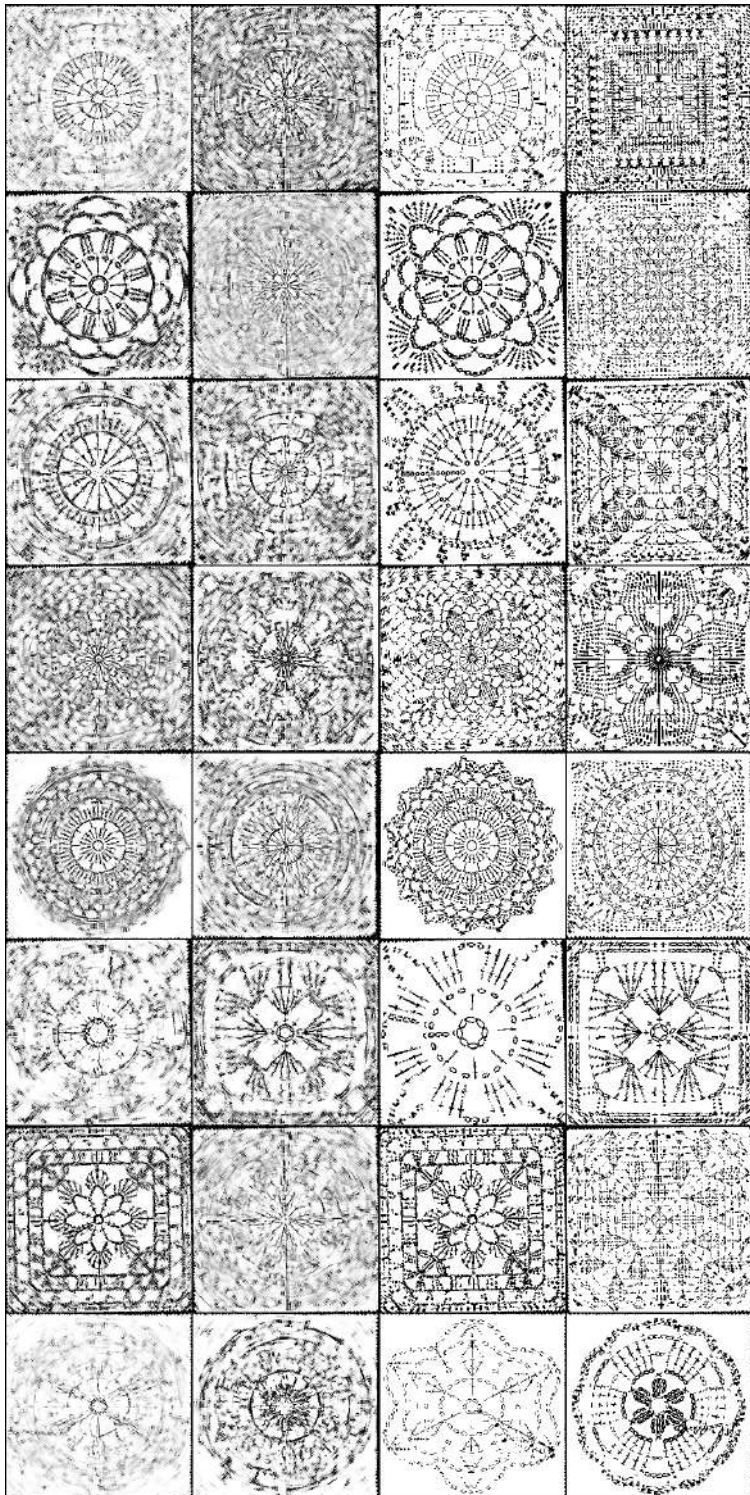
**Rysunek 1.4.** Schematy szydełkowe (dwa górne rzędy) i ich rekonstrukcje (dwa dolne rzędy) przygotowane przez sieć z dekoderem Bernoulliego

## **Dodatek 2. Wyniki eksperymentu z grafikami w układzie współrzędnych biegunowych**

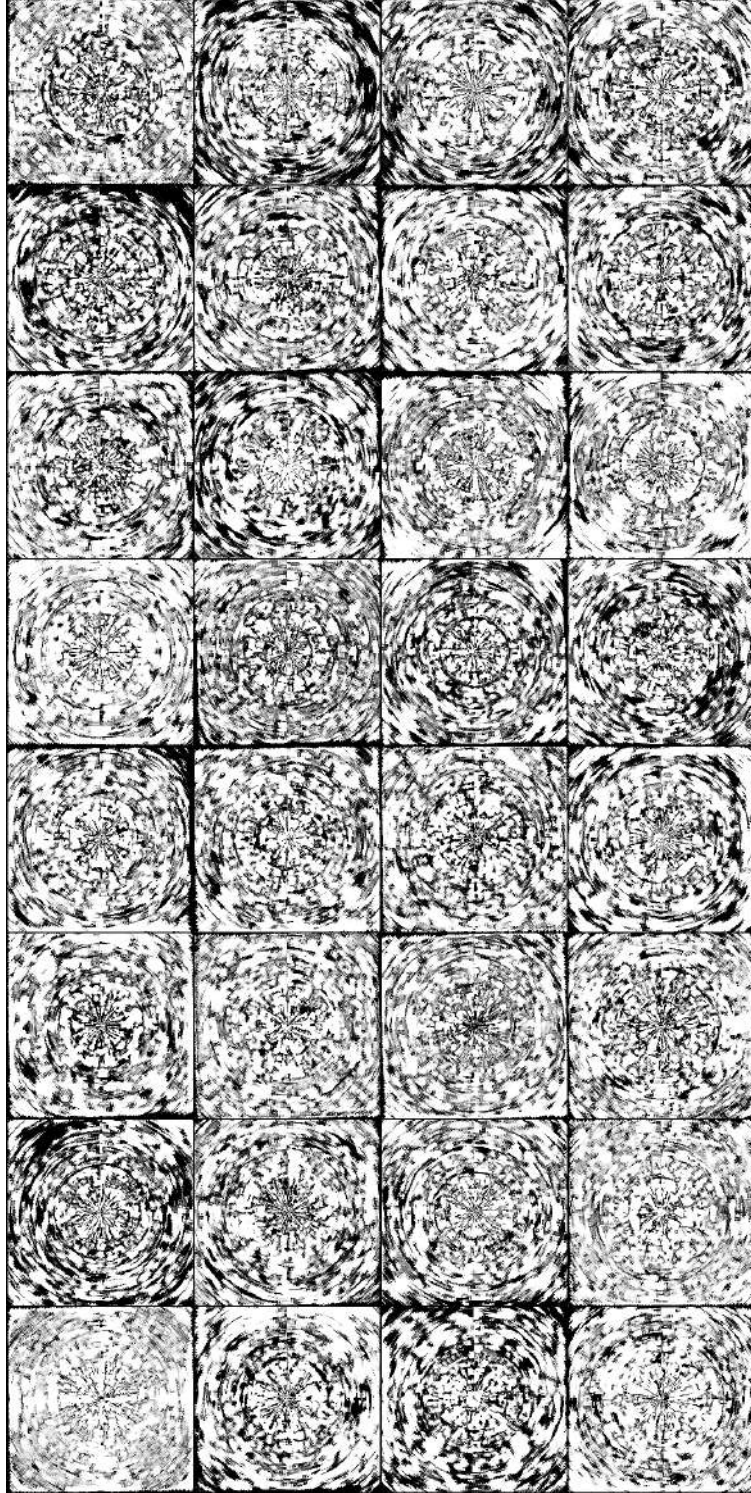


**Rysunek 2.1.** Schematy szydełkowe wygenerowane przez sieć z dekodorem gaussowskim



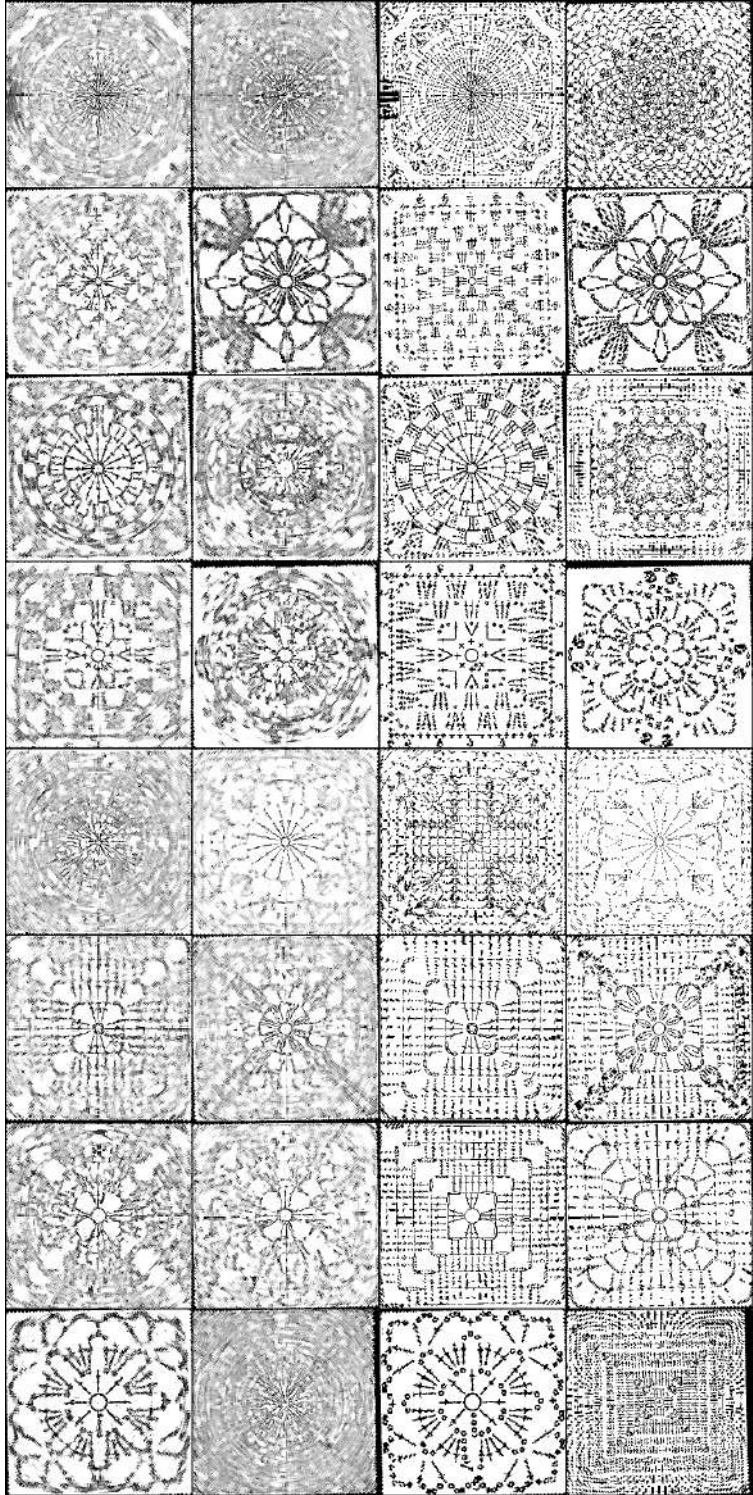


**Rysunek 2.2.** Schematy szydełkowe (dwa górne rzędy) i ich rekonstrukcje (dwa dolne rzędy) przygotowane przez sieć z dekodernem gaussowskim



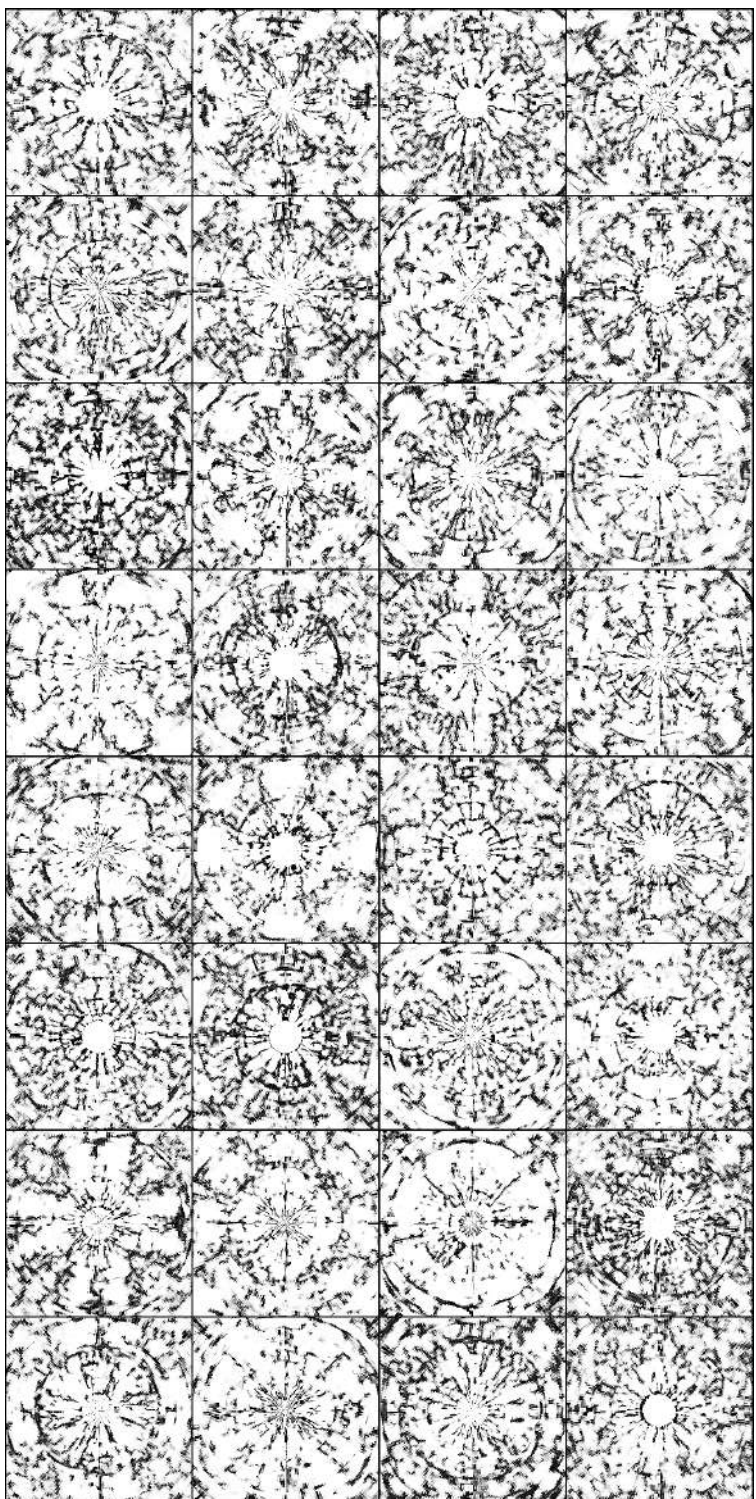
**Rysunek 2.3.** Schematy szydełkowe wygenerowane przez sieć z dekodorem Bernoulliego



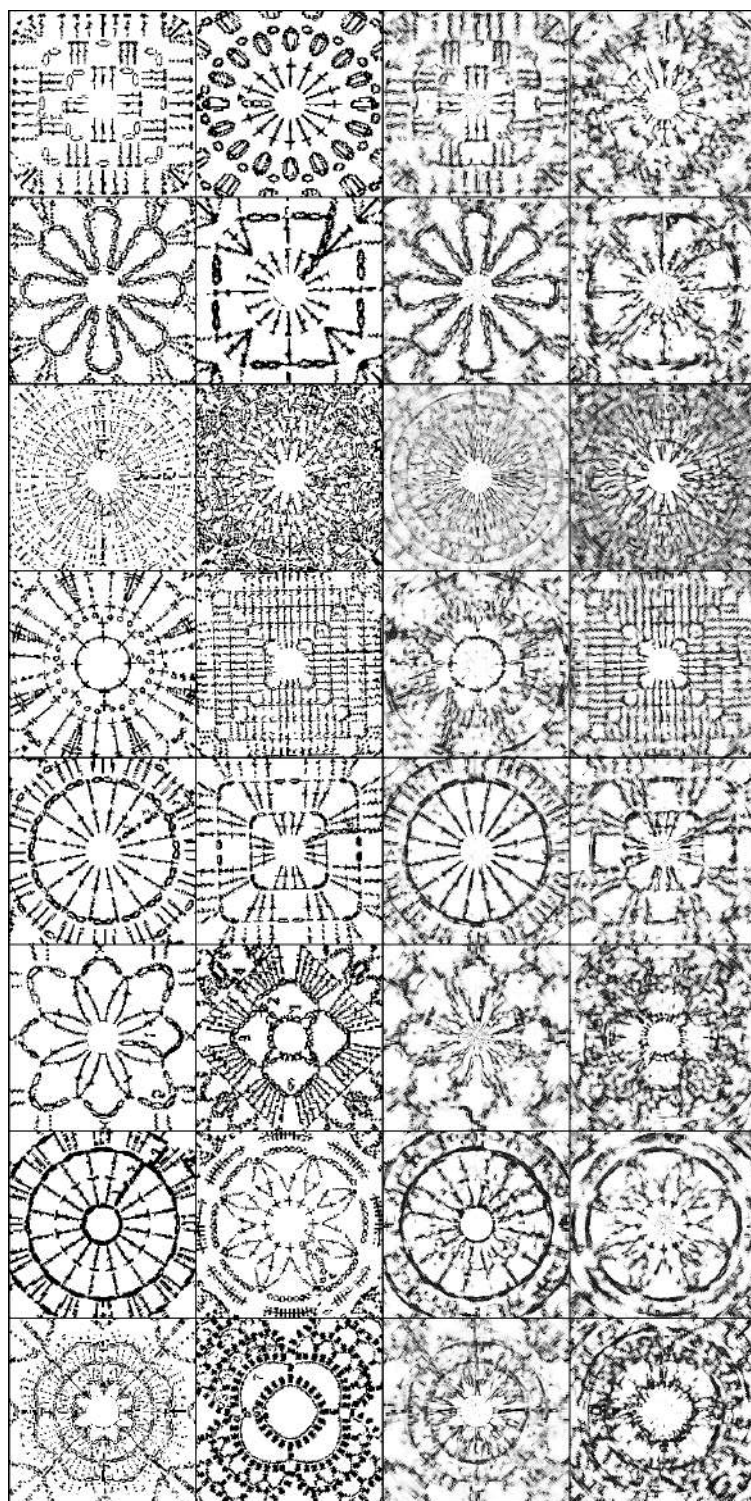


**Rysunek 2.4.** Schematy sztydkowe (dwa górne rzędy) i ich rekonstrukcje (dwa dolne rzędy) przygotowane przez sieć z dekodern Bernoulliego

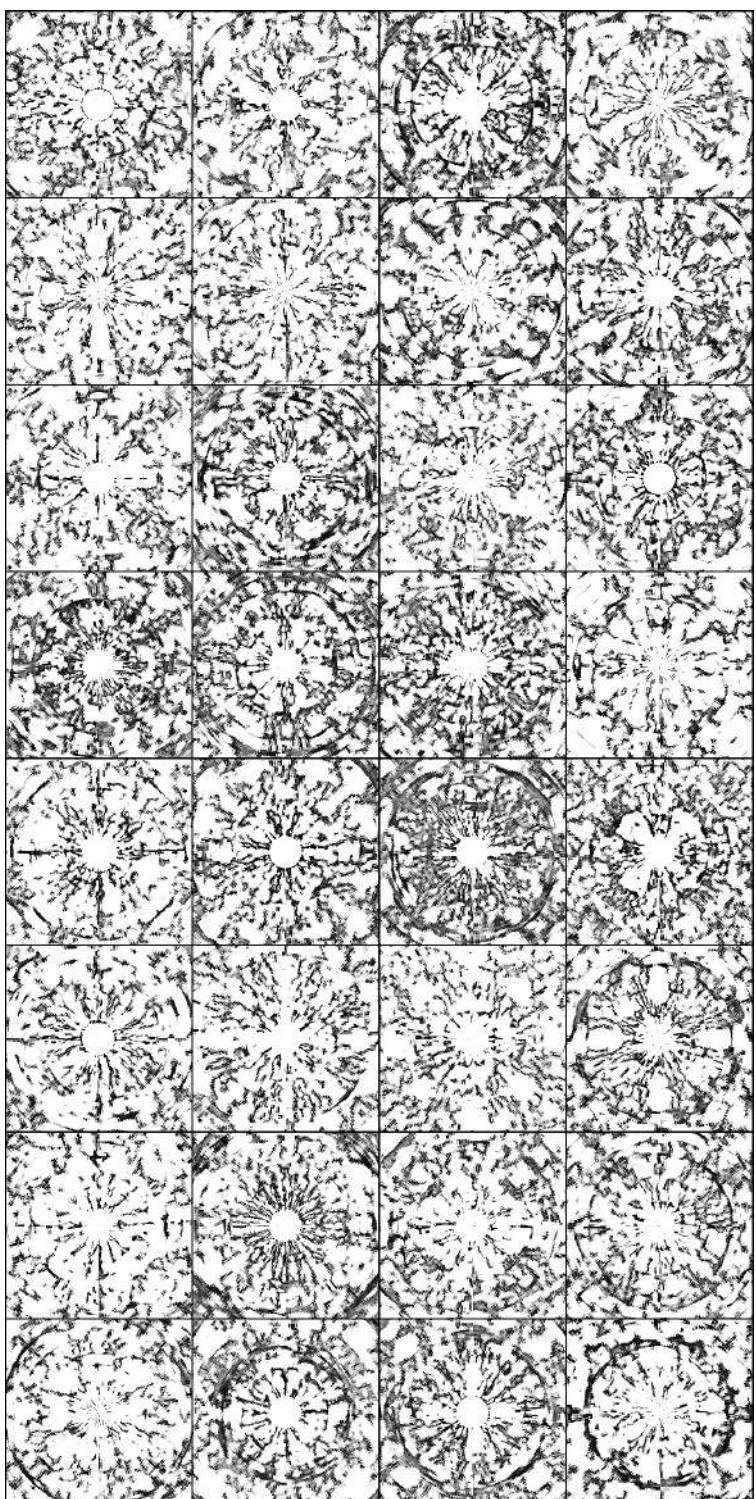
### **Dodatek 3. Eksperyment z pominięciem niekorzystnych zjawisk w układzie współrzędnych biegunowych**



**Rysunek 3.1.** Wycinki pierścieniowe schematów sztydkowych wygenerowanych przez sieć z dekodern gaussowskim

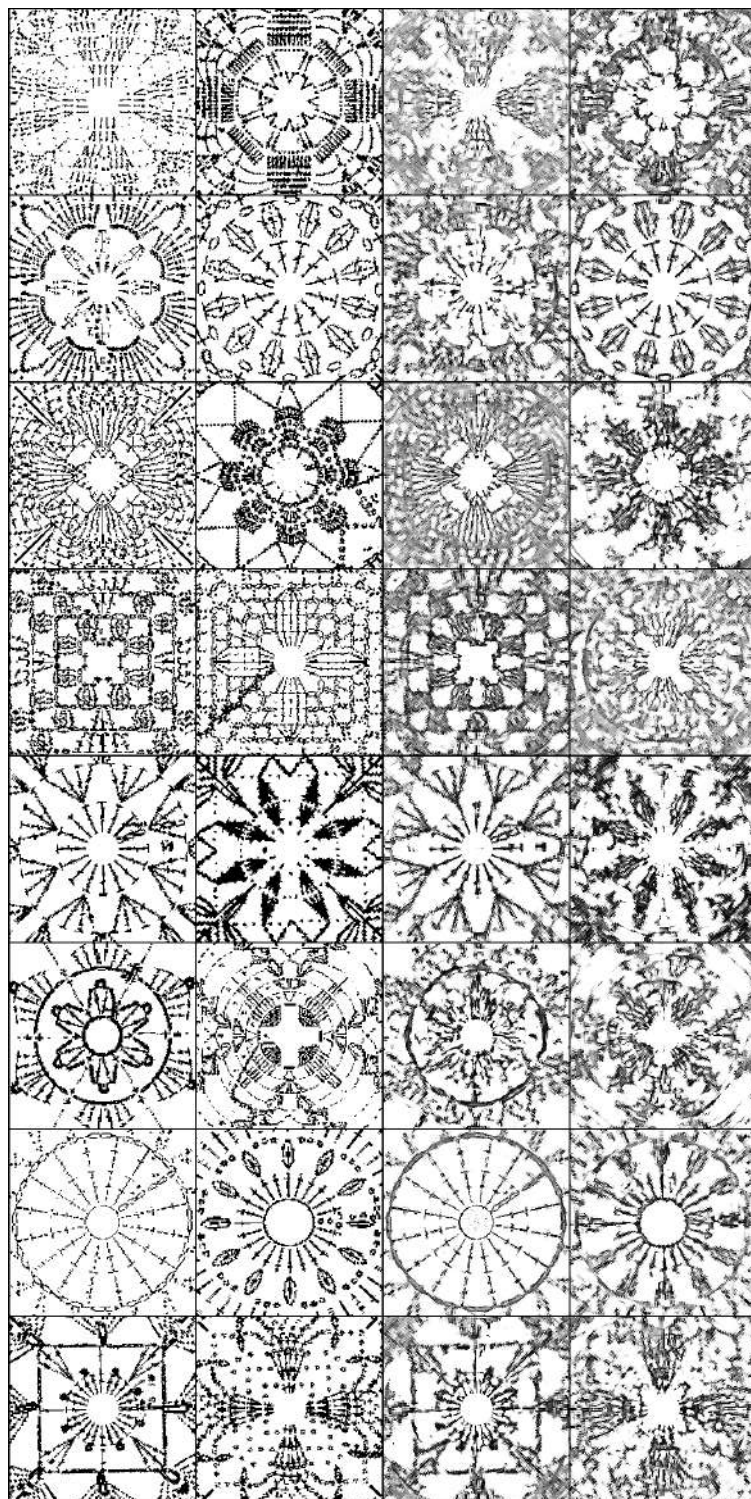


**Rysunek 3.2.** Wycinki pierścieniowe schematów szydełkowych (dwa górne rzędy) i ich rekonstrukcje (dwa dolne rzędy) przygotowane przez sieć z dekodowaniem gaussowskim



**Rysunek 3.3.** Wycinki pierścieniowe schematów szydełkowych wygenerowanych przez sieć z dekodorem Bernoulliego

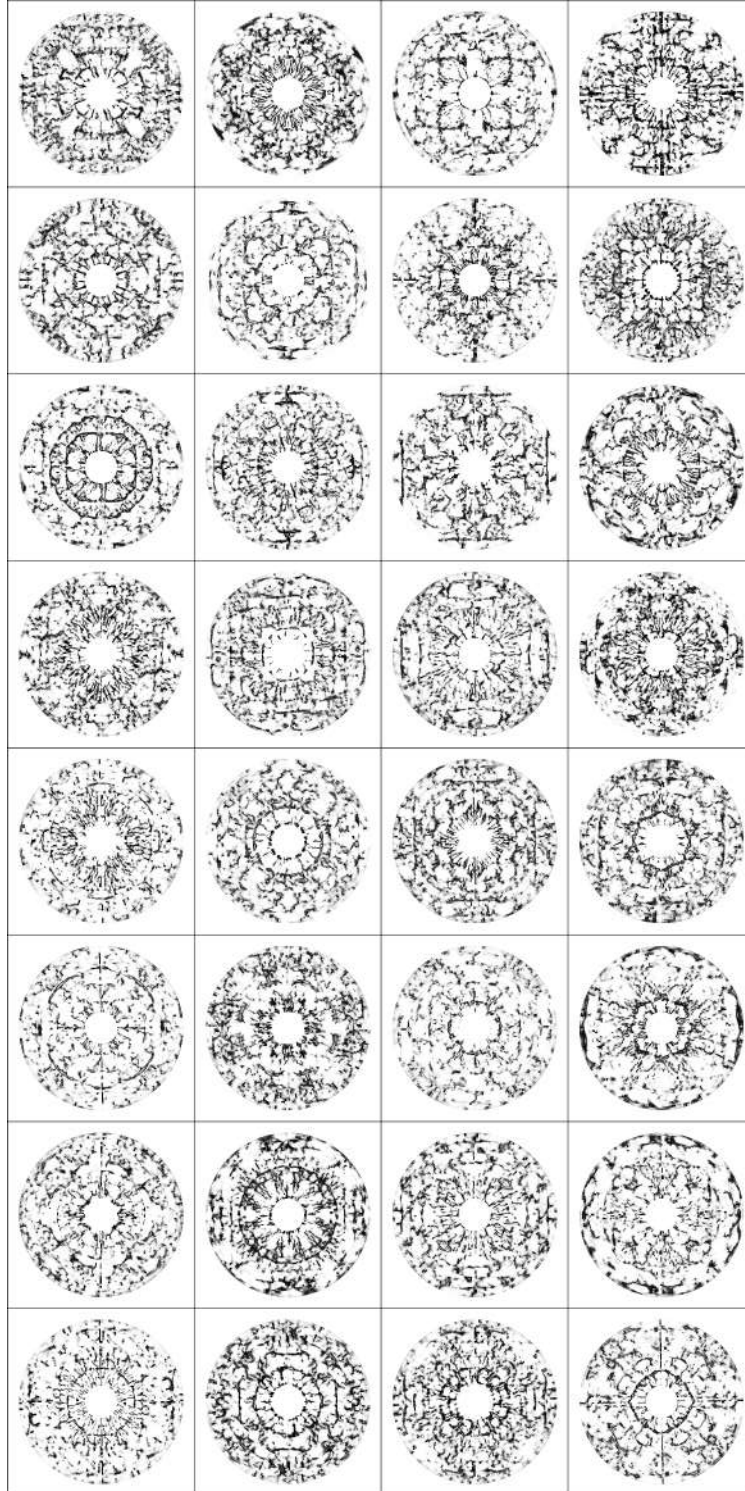




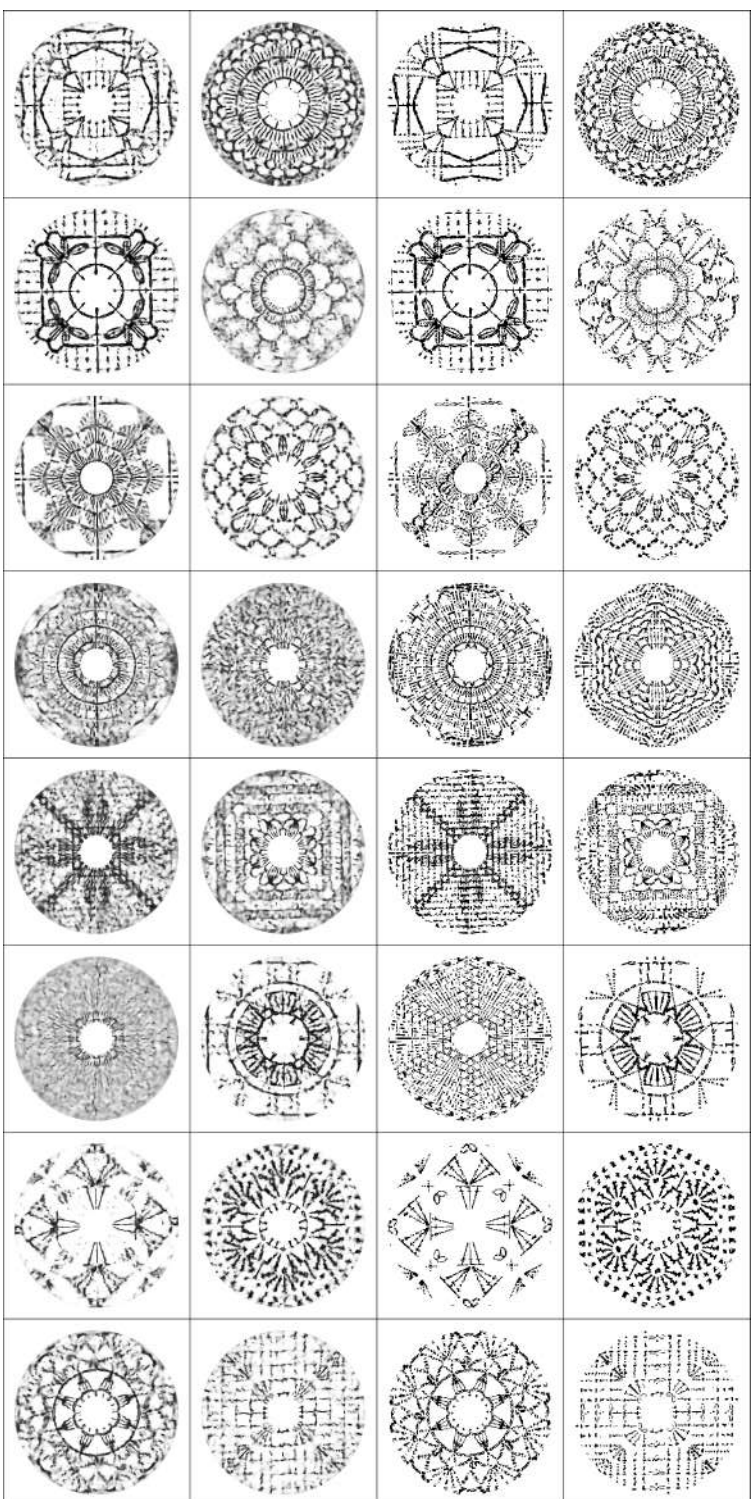
**Rysunek 3.4.** Wycinki pierścieniowe schematów szydełkowych (dwa górne rzędy) i ich rekonstrukcje (dwa dolne rzędy) przygotowane przez sieć z dekodorem Bernoulliego

## **Dodatek 4. Wyniki eksperymentu z kawałkami grafik w układzie współrzędnych biegunowych**

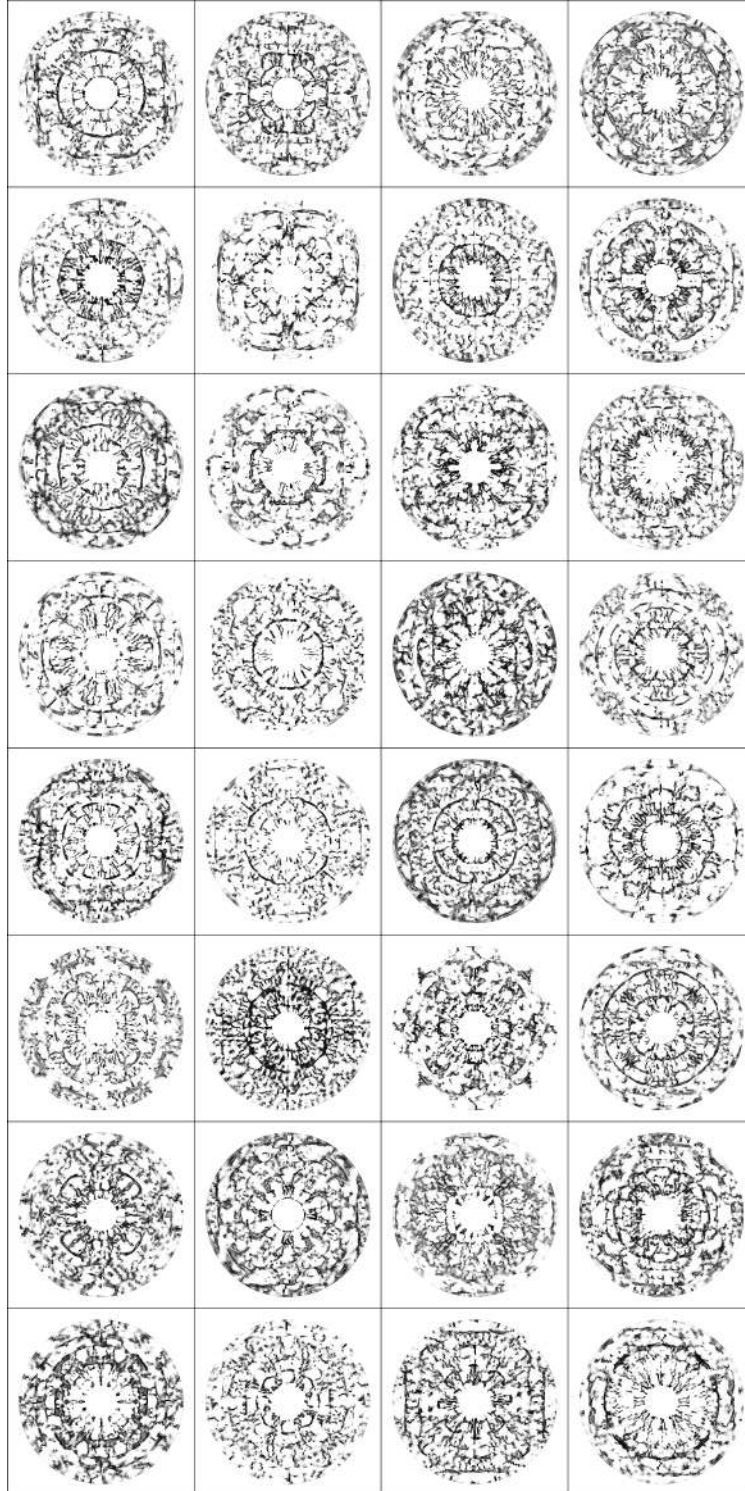




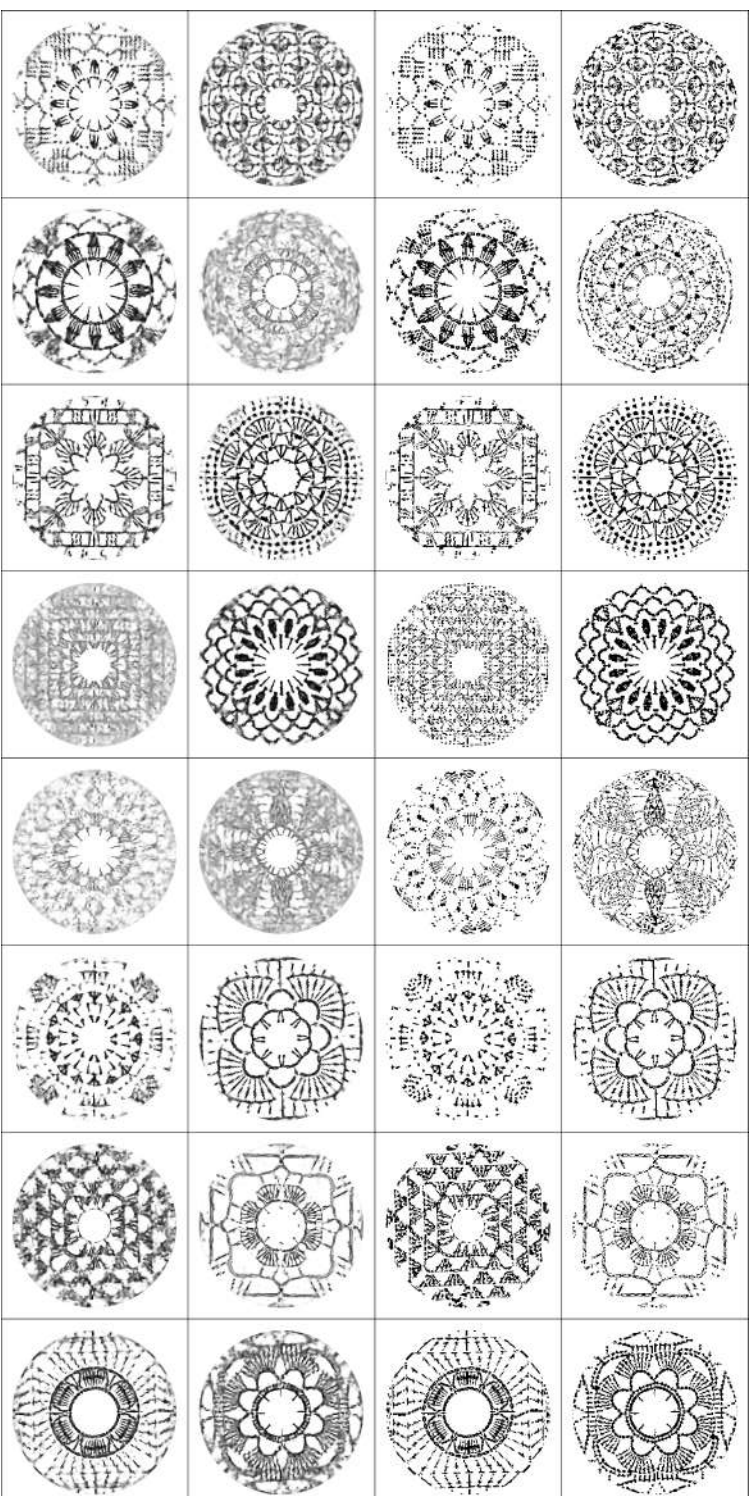
**Rysunek 4.1.** Część informacji o schemacie szydełkowym wygenerowana przez sieć z dekodorem gaussowskim



**Rysunek 4.2.** Części informacji o schemacie szydełkowym (dwa górne rzędy) i ich rekonstrukcje (dwa dolne rzędy) przygotowane przez sieć z dekodерem gaussowskim



**Rysunek 4.3.** Część informacji o schemacie szydełkowym wygenerowana przez sieć z dekodernem Bernoulliego



**Rysunek 4.4.** Części informacji o schemacie szydełkowym (dwa górne rzędy) i ich rekonstrukcje (dwa dolne rzędy) przygotowane przez sieć z dekodерem Bernoulliego