Rozpoznawanie Gatunków Muzycznych za Pomocą Sieci Neuronowych oraz Rozszerzenie Danych Treningowych o Autorskie Utwory EDM (Raport Badawczy)

Maciej Wojciechowski, Uniwersytet Gdański, wydział Matematyki, Fizyki i Informatyki

Czerwiec 2025, Informatyka (profil praktyczny) — sem. 4, Inteligencja Obliczeniowa

Cyfrowe rozpoznawanie gatunków muzycznych jest zadaniem prezentującym wiele przeszkód — od reprezentacji złożonych fal dźwiękowych w środowisku binarnym, po wybór odpowiedniego fragmentu utworu do celów analitycznych. W tym przypadku należy również szczególnie uwzględnić etykę pozyskiwania danych źródłowych, gdyż autorskie prawa majątkowe w Polsce nie ulegają przedawnieniu przed upływem 70 lat od śmierci autora. W konsekwencji, znacznemu utrudnieniu podlega pozyskanie wystarczającej ilości danych treningowych reprezentujących najnowsze gatunki muzyczne (EDM, trap, itd.).

Dopiero w konsekwencji zmierzenia się z wyżej wymienionymi przeszkodami możliwym staje się dobranie odpowiednich technik uczenia maszynowego, generalizacji oraz augmentacji danych, jaki i ich optymalnej parametryzacji.

Wstęp

Kontekst i motywacja

Klasyfikacja gatunków muzycznych jest obszernym tematem dotykających wielu dziedzin nauki, biznesu oraz sztuki. Serwisy streamingowe udostępniające ogromne bazy utworów muzycznych użytkownikom końcowym stają przed następującymi wyzwaniami ściśle powiązanymi z klasyfikacją gatunków publikowanych utworów:

- dostarczenie użytkownikom plików dźwiękowych od obserwowanych przez nich artystów
- proponowanie podobnych utworów użytkownikom (w celu przedłużenia czasu korzystania z serwisu)
- generowanie zarobku rozpraszanego między serwis, niezależnych artystów oraz (bezpośrednio związanych kontraktami) wytwórni muzycznych.

Producenci muzyczni oraz inżynierowie dźwięku – w szczególności inżynierowie miksu (ang. „mix engineer") oraz wykończenia (ang. „mastering engineer") – w celu wytworzenia gotowego produktu wykorzystują programy komputerowe generujące dźwięk lub modyfikujące istniejące ścieżki dźwiękowe. Te programy w środowisku producentów nazywane są potocznie „wtyczkami" (ang. „plugin").

Jedną z firm korzystających z dynamicznej klasyfikacji gatunków muzycznych jest Sonible (https://www.sonible.com). Sonible wytwarza oprogramowanie oparte na uczeniu maszynowym wspomagające producentów muzycznych sugestiami modyfikacji dźwięku reagującymi na zmiany „na żywo". Aby uzyskać zamierzony efekt, wtyczki tej firmy klasyfikują gatunek utworu, a następnie dobierają optymalne parametry korekcji lub uwydatniające pojedyncze ścieżki lub pełne utwory.

Sonible, o swoim produkcie Smart:EQ 4:

> *The centerpiece of smart:EQ 4 is the smart:filter — it automatically balances the signal based on a target Profile you choose for your tracks. The smart:filter within each EQ instance comes with a range of features to adjust the AI-powered processing to your liking.*

To „zaszufladkowanie" jest konieczne, gdyż każdy gatunek muzyczny ma inne standardy techniczne. Ilustracyjnie:

- głośność i dynamika
- balans spektrum częstotliwości (np. bas vs tony wysokie)
- dostosowanie do medium (EDM grany jest zwyczajowo jednokanałowo „mono" w klubach na profesjonalnych kolumnach, Pop często wielokanałowo „stereo" na głośnikach konsumenckich lub wbudowanych)

W przypadku muzyki elektronicznej, legalne pozyskanie danych treningowych wiąże się z masowymi wydatkami odkupowania od wytwórni muzycznych praw do wykorzystania utworów nadal objętych prawami autorskimi. Jest to powodem słabej reprezentacji (lub jej całkowitego braku) w zestawach danych lub algorytmach klasyfikacji. Dla przykładu: GTZAN nie posiadał

żadnych danych z gatunku „elektroniczna" po roku ~1980, co kompletnie omija Progressive House, Future Bass oraz inne, które z pewnością klasyfikował jako Pop[1].

Szybszy podział ogromnych zestawień utworów może również znacząco przyspieszyć rozwój rozwiązań generujących utwory muzyczne na podstawie promptów (np. https://suno.com).

Jak osiągnąć taką klasyfikację stosunkowo nieskomplikowanym programem komputerowym wymagającym niewielkich zasobów hardware? Jaki wpływ na wyniki ma uzupełnienie danych treningowych o dodatkową kategorię „EDM" (kategoria ta została w zupełności wypełniona autorskimi utworami wymienionymi w ostatniej sekcji niniejszego raportu)?

Historyczna klasyfikacja utworów

Przed szerokim wprowadzeniem podejścia uczenia maszynowego, klasyfikacja gatunków utworów była naiwna i prymitywna. Brano pod uwagę głównie:

- Tempo (w/w Progressive House ma stałe tempo 128 BPM)
- Długość trwania (utwory muzyki klasycznej potrafią trwać 10+ min, natomiast rapowe nierzadko ~1.5 min)
- Tonację (dużo gatunków sugeruje użycie konkretnej tonacji)

Dziś powszechnym stało się podejście generalizacji danych w oparciu o uczenie maszynowe oraz sieci neuronowe. Prężnie rozwijająca się w tym kierunku technologia umożliwia wytrenowanie prostych sieci w warunkach amatorskich, wykorzystując równoważność zasobów dostępnych nawet w niektórych telefonach komórkowych (flagowce firm Apple oraz Samsung oferują systemowo lokalne funkcje AI — bez konieczności połączenia z chmurą).

Przygotowanie danych treningowych

Szukanie charakterystyk w pojedynczej (mono) złożonej fali dźwiękowej wykazałoby niezwykle niską skuteczność. W tym przypadku rozwiązaniem jest powszechnie używana w muzyce technika rozwarstwienia złożonej fali na jej składowe — Fast Fourier Transform (FFT). Ta transformacja umożliwia graficzną wizualizację dźwięku oraz przekazanie modelowi dane o stosunku amplitud poszczególnych fal oraz relatywnemu zagęszczeniu spektrum (muzyka elektroniczna powszechnie wypełnia spektrum szumem, klasyczna ma dużo „luk").

Można dodatkowo uprościć model załamaniem do mono — zachowaniem jedynie informacji, które nie są powielane lub wzajemnie wykluczane w kanałach lewo/prawo.

Kolejną techniką umożliwiającą dekomplikację modelu jest zmiana częstotliwości próbkowania. Jest to miara (Hz) ile razy na sekundę pobierana jest aktualna amplituda fali. Mniejsza częstotliwość oznacza niemożliwość zapisu skrajnie wysokich częstotliwości, natomiast nie ma żadnego wpływu na częstotliwości niższe niż połowa częstotliwości próbkowania (dla standardowego 44.1kHz można perfekcyjnie zapisać częstotliwości do 22 050Hz. Zdrowy młody dorosły słyszy maksymalnie ~19kHz). Natomiast za rozpoznawanie mowy odpowiedzialny jest

[1] Najlepszy model wytrenowany na nieaugmentowanych danych GTZAN dla piosenki EDM „ILLENIUM — Crawl Outta Love" wygenerował wynik: „Pop (63%)".

zakres ~2kHz–5kHz. Można zatem założyć, że dane zapisane w skrajnie górnej części spektrum są głównie małowartościowym „syczeniem".

Należy zwrócić uwagę na poprawną konwersję częstotliwości próbkowania, gdyż mechaniczne wyrzucenie danych z pliku doprowadzi do zniekształceń oraz odbić (ang. aliasing). Aby tego uniknąć, należy najpierw zaaplikować filtr dolnoprzepustowy, aby pozbyć się za wysokich częstotliwości i zminimalizować te zniekształcenia.

Utrudnione pozyskanie szerokiej gamy danych prezentuje konieczność ich augmentacji. Aby pozyskać większą ilość plików dźwiękowych można dokonać modyfikacji fal w nich zapisanych:

- Transpozycja
- Zmiana tempa
- Podział plików na mniejsze fragmenty
- Zaszumianie
- (ostrożna) korekcja (EQ)[2]
- (ostrożna) kompresja lub ekspansja dynamiki[3]

Poniższa analiza została przeprowadzona z wykorzystaniem zestawu danych treningowych GTZAN, zawierającego 1000 plików (*.wav 16-bit mono 22050Hz) podzielonych na kategorie — po 100 plików z 10 gatunków.

Dane zostały również wzbogacone o dodatkowy folder 52 oryginalnych plików zawierających fragmenty autorskich utworów. Augmentacja doprowadziła do rozszerzenia tego zestawu do 208 plików przewyższając dwukrotnie ilość plików w folderach GTZAN. Mniejsza różnorodność (wszystkie utwory wyprodukowane przez jednego producenta) okazały się nie przesłonić skuteczności klasyfikacji.

Architektura modelu

Warstwa wejścia

Akceptuje tensor spektrogramu mel (128 pasm mel, T jednostek czasu, 1 kanał)

Warstwa konwolucyjna

Sekwencja *n* bloków Conv2D, gdzie *n = len(filters)*

Każdy blok składa się z następujących elementów:

- Conv2D(f, (3×3), activation=<chosen>, padding='same')

[2] Rożne gatunki mają różny balans częstotliwości. Można przykładowo zaaplikować tzw. „all-pass filter", który zmienia fazę fali, bez modyfikacji balansu.
[3] Zbyt agresywna kompresja zbliży plik do standardów muzyki elektronicznej, natomiast zbyt agresywna ekspansja doprowadzi do uwydatnienia charakterystyk znanych muzyce Jazz.

- BatchNormalization()
- MaxPool2D((2×2))
- Dropout(rate=dropouts[i])

Warstwa dense “head”

- Flatten()
- Dense(256, activation=<chosen>)
- BatchNormalization()
- Dropout(rate=dropouts[-1])
- Dense(num_classes, activation='softmax')

Hiperparametry

- Aktywacja: {relu, tanh, elu, selu, gelu}
- Zestawy filtrów: [[16,32,64], [32,64,128], [64,128,256]]
- Profile “dropout” np. [0.25,0.25,0.25,0.5]

Trenowanie

- Data streaming
    - Dwa generator MelDataset (trening vs. walidacja), batch_size ∈ {8,16,32}
- Loss & optimiser
    - loss='sparse_categorical_crossentropy'
    - optimizer=Adam(lr=<swept value>) (default 1e-4)
- Class balancing
    - Wyliczenie wag poprzez sklearn.utils.compute_class_weight('balanced', classes=...)
- Macierz hiperparametrów
    - Epochs: {8,15,30}
    - Batch size: {8,16,32}
    - Activation, filters, dropout as above
    - (Opcjonalny) learning-rate sweep

Strategia ewaluacji

Monitorowanie *val_loss* i *val_accuracy*, inspekcja macierzy błędów.

Metryki

- Accuracy
- Loss
- Macierze błędów

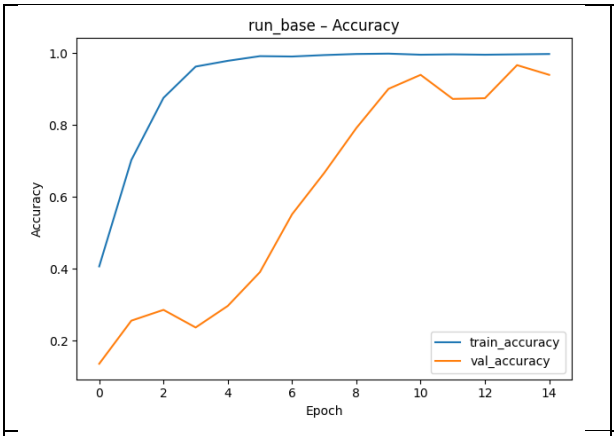
Testy dodatkowe

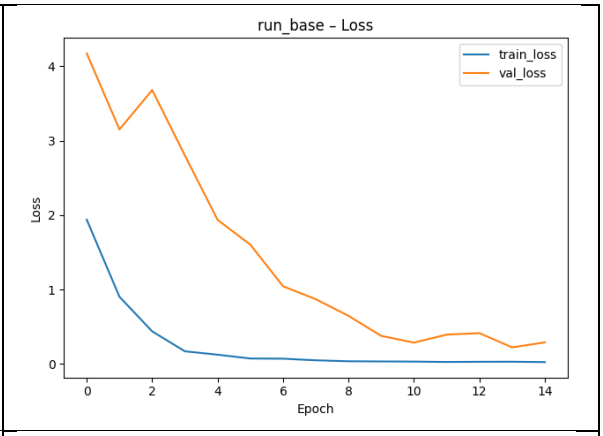
Ewaluacja działania programu na popularnych utworach kultury masowej oraz ręczne porównanie wyniku z rzeczywistą klasyfikacją pliku. Testy empiryczne pozwalają zweryfikować i analizować znalezione błędy w klasyfikacji na konkretnych przykładach.

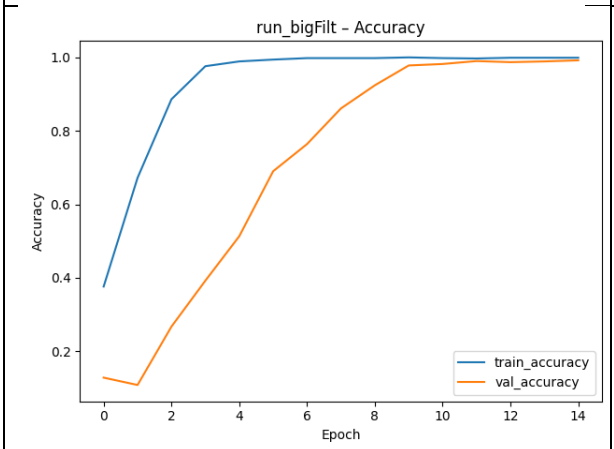
Wyniki testów

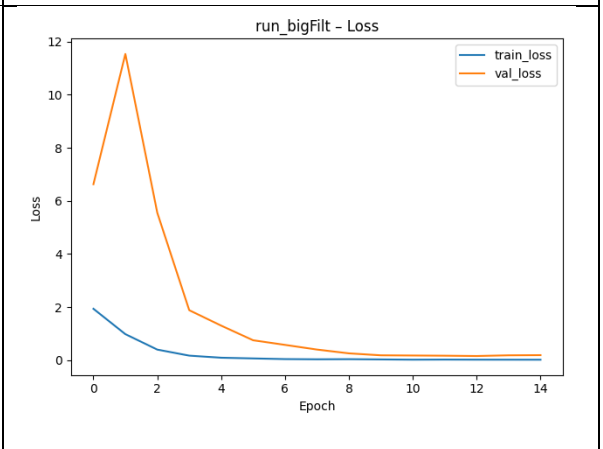
Wykresy accuracy oraz loss dla modeli z następującymi parametrami:

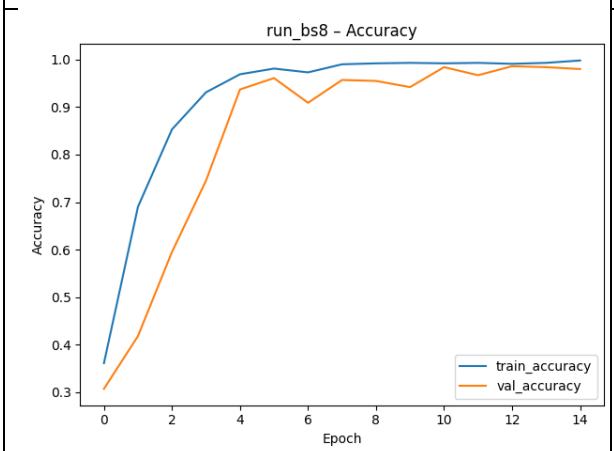
name	batch_size	epochs	activation	filters	dropouts
run_base	16	15	relu		
run_relu	16	15	relu		
run_tanh	16	15	tanh		
run_elu	16	15	elu		
run_selu	16	15	selu		
run_gelu	16	15	gelu		
run_bs8	8	15	relu		
run_bs32	32	15	relu		
run_smallFilt	16	15	relu	16;32;64	
run_bigFilt	16	15	relu	64;128;256	
run_dropHigh	16	15	relu		0.3;0.3;0.3;0.6
run_best_test	8	8	selu	16;32;64	0.3;0.3;0.3;0.6

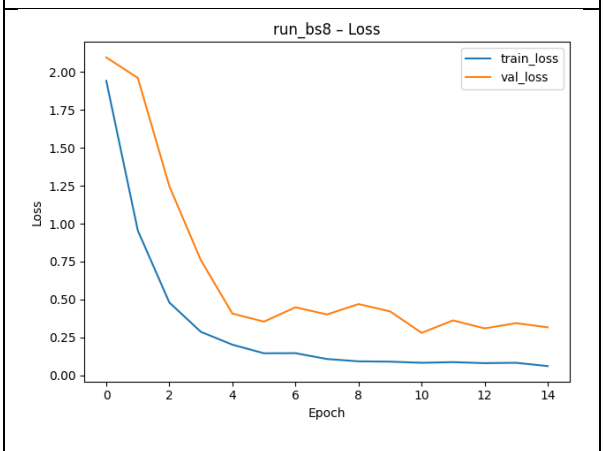

run_base – Accuracy
Accuracy
Epoch
train_accuracy
val_accuracy

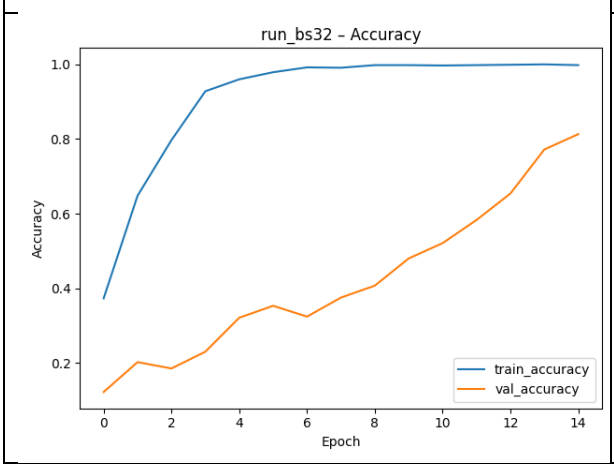

run_base – Loss
Loss
Epoch
train_loss
val_loss

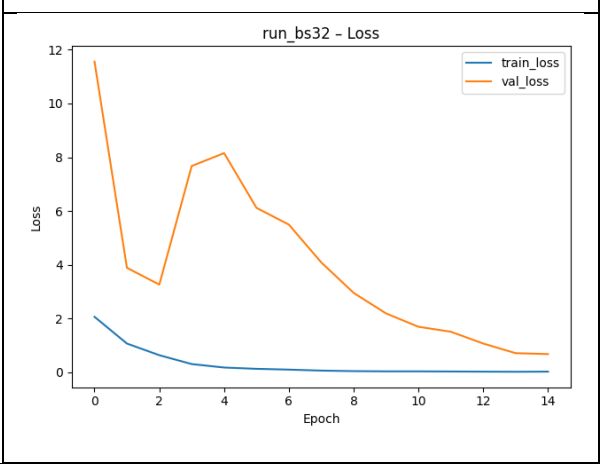

run_bigFilt – Accuracy
Accuracy
Epoch
train_accuracy
val_accuracy

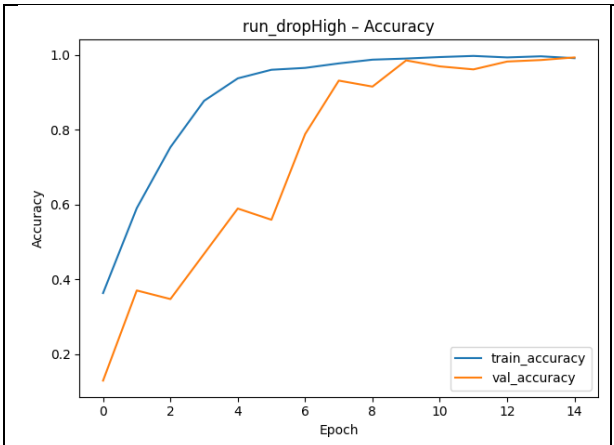

run_bigFilt – Loss
Loss
Epoch
train_loss
val_loss

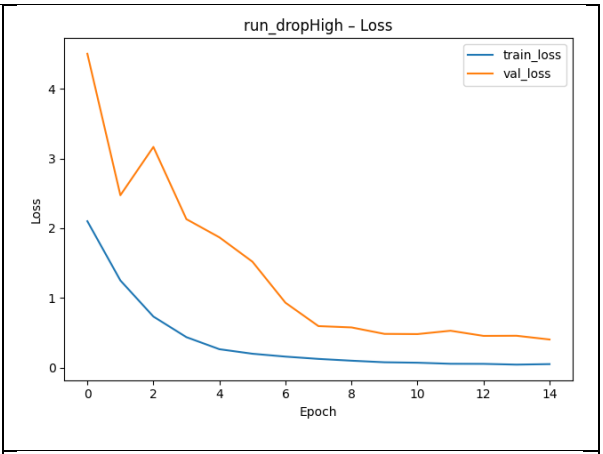

run_bs8 – Accuracy
Accuracy
Epoch
train_accuracy
val_accuracy

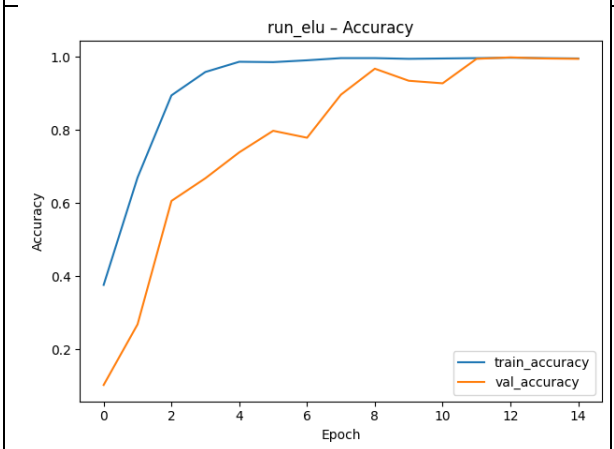

run_bs8 – Loss
Loss
Epoch
train_loss
val_loss

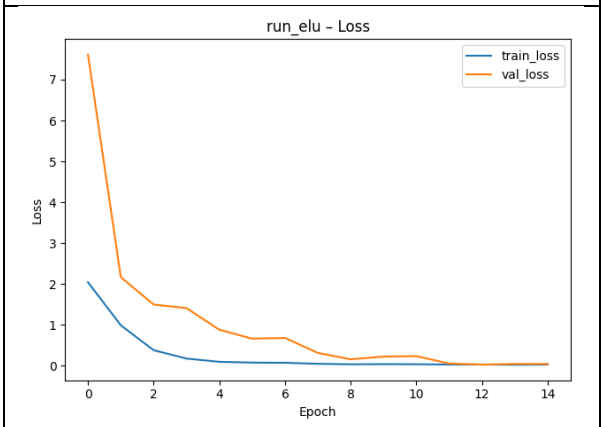

run_bs32 – Accuracy
Accuracy
Epoch
train_accuracy
val_accuracy

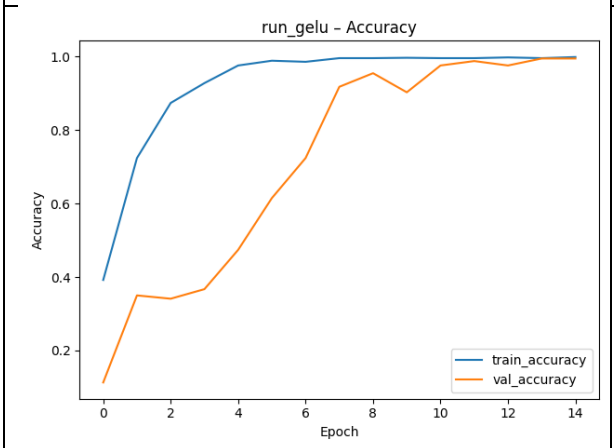

run_bs32 – Loss
Loss
Epoch
train_loss
val_loss

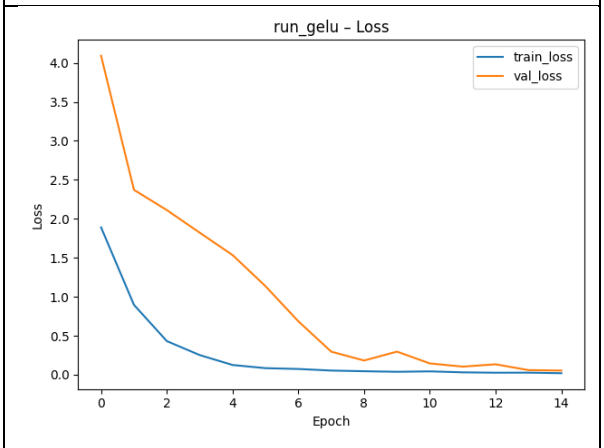

run_dropHigh – Accuracy
Accuracy
Epoch
train_accuracy
val_accuracy

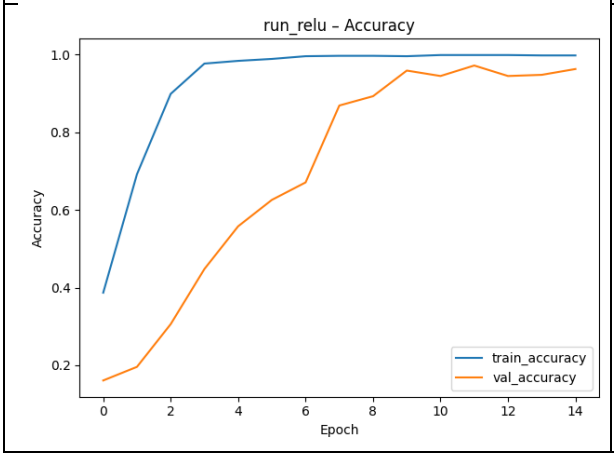

run_dropHigh – Loss
Loss
Epoch
train_loss
val_loss

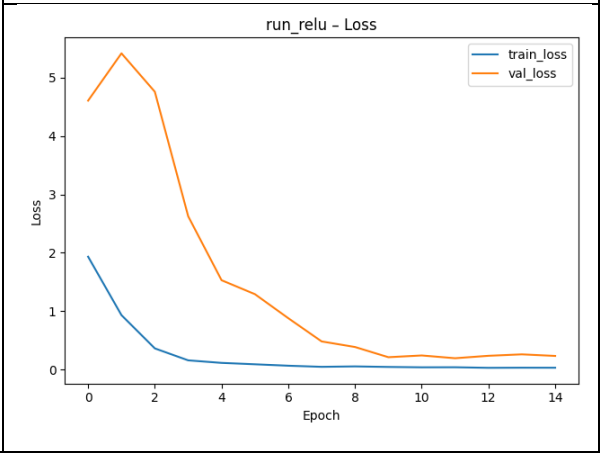

run_elu – Accuracy
Accuracy
Epoch
train_accuracy
val_accuracy

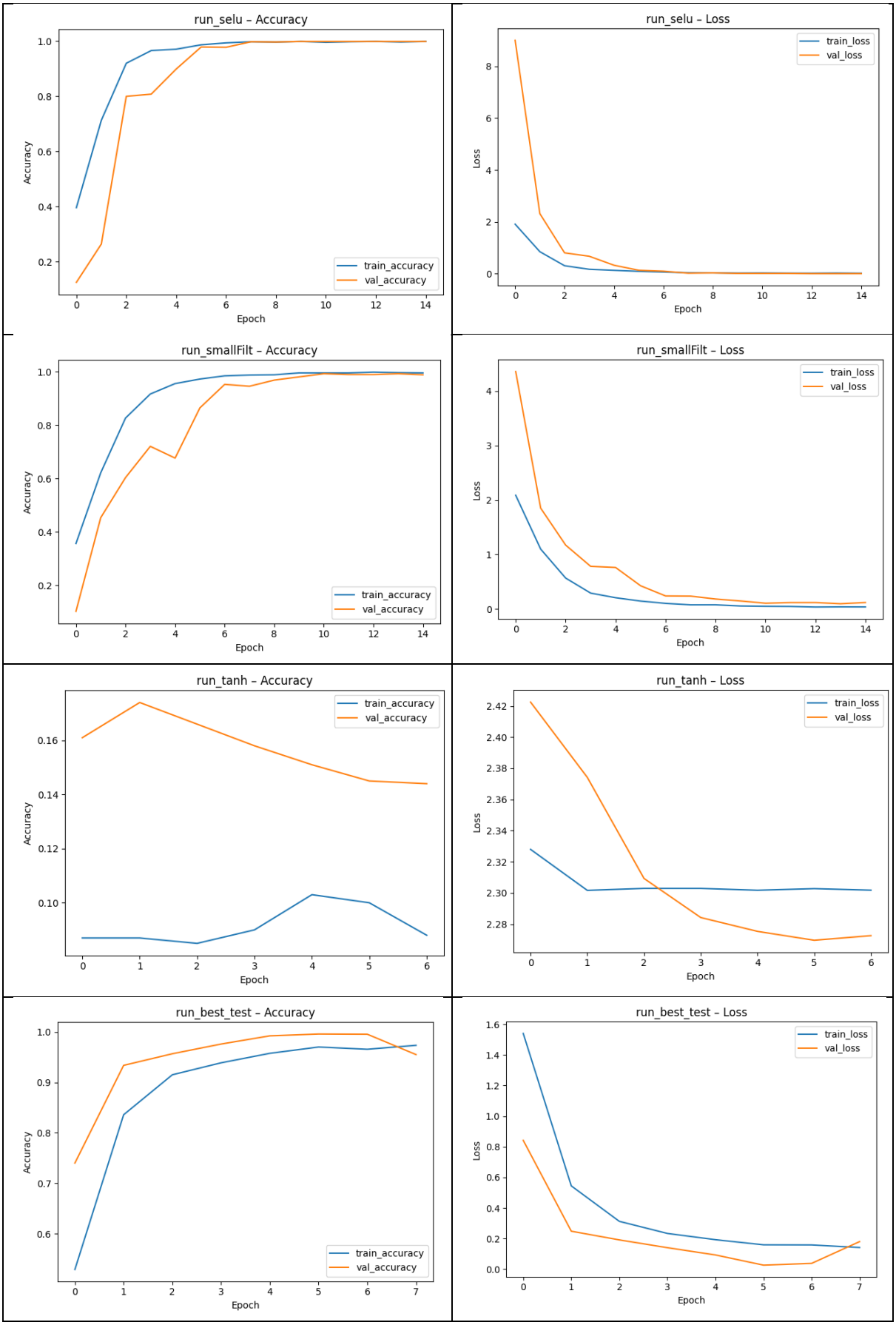

run_elu – Loss
Loss
Epoch
train_loss
val_loss


run_gelu – Accuracy
Accuracy
Epoch
train_accuracy
val_accuracy


run_gelu – Loss
Loss
Epoch
train_loss
val_loss


run_relu – Accuracy
Accuracy
Epoch
train_accuracy
val_accuracy


run_relu – Loss
Loss
Epoch
train_loss
val_loss

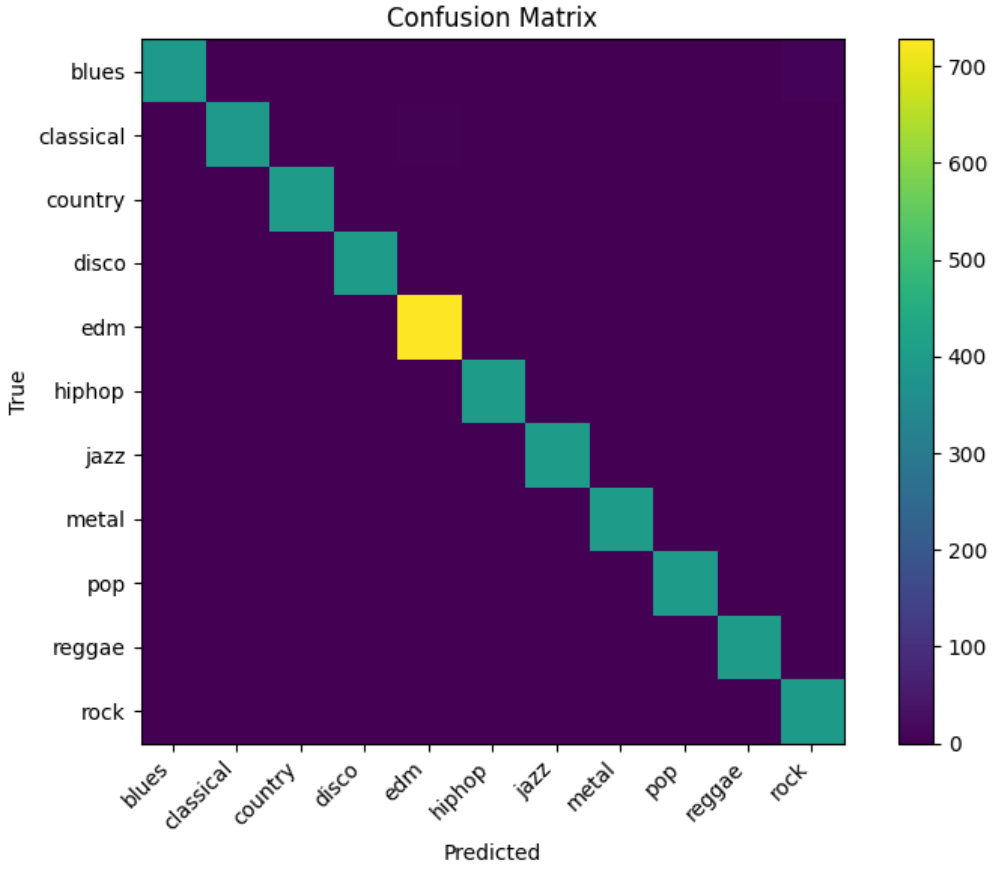

run_selu – Accuracy
Accuracy
Epoch
train_accuracy
val_accuracy
run_selu – Loss
Loss
Epoch
train_loss
val_loss
run_smallFilt – Accuracy
Accuracy
Epoch
train_accuracy
val_accuracy
run_smallFilt – Loss
Loss
Epoch
train_loss
val_loss
run_tanh – Accuracy
Accuracy
Epoch
train_accuracy
val_accuracy
run_tanh – Loss
Loss
Epoch
train_loss
val_loss
run_best_test – Accuracy
Accuracy
Epoch
train_accuracy
val_accuracy
run_best_test – Loss
Loss
Epoch
train_loss
val_loss

Powyższe wykresy przedstawiają szybkość i jakość uczenia z różnymi parametrami. Ostatni zestaw hiperparametrów — run_best_test — przedstawia trening ostatecznego modelu.

Wszystkie konfiguracje (za wyjątkiem aktywacji tanh lub batch_size=32) wykazały skuteczność >90% przed epoką 9. Ostateczny model osiągnął ten próg w epoce 2. W nim, końcowy spadek *accuracy* oraz wzrost *loss* pozwala stwierdzić, że dalszy trening na tych danych nie przyniósłby poprawy wydajności.

W celu usprawnienia modelu jedynym wyjściem zdaje się powiększenie zestawu treningowego o kolejne kilkaset, lub nawet kilkanaście tysięcy utworów równomiernie rozdzielonych po kategoriach gatunku.

Dla ostatecznego modelu, macierz błędów wskazuje na znakomite nauczenie się charakterystyk gatunków, z wyraźnym przeuczeniem w kategorii EDM — zapewne z powodu mniej różnorodnych danych pochodzących z jednego źródła oraz ich mniejszej liczebności przed augmentacją:



Testy z pomocą aplikacji konsolowej

Dla testów „real-world" — na utworach popularnych artystów wyniki są zadowalające, natomiast przeszkodą okazał się odpowiedni wybór fragmentu piosenki w celu kategoryzacji gatunku. Zwrotki wielu gatunków są łudząco podobne. Bez kontekstu refrenu, nawet ocena z wykorzystaniem losowej próbki grupy ochotników zapewne wykazałaby korelację klasyfikacji z gustem muzycznym uczestników.

W tym celu program pobiera 3 30-sekundowe próbki (środek oraz jego bezpośredni sąsiedzi). W przypadku wejścia krótszego niż 90 sekund, program zapętla wejście. Następnie generuje 3 predykcje uporządkowane względem pewności, malejąco.

Na skutek tego uporządkowania, program otrzymał tzw. „emergent quality" — fragment z najpewniejszą predykcją jest również tym, który zawiera największą część refrenu (lub dropu).

Song	*Genre*	*Predictions*
Said The Sky — Legacy	EDM	After: edm (97.1% confidence) Before: edm (94.6% confidence) Middle: edm (94.0% confidence)
Said The Sky — Reminisce	Pop-EDM	After: edm (91.5% confidence) Before: edm (86.2% confidence) Middle: pop (61.3% confidence)
Gryffin, Illenium — Feel Good	EDM	After: edm (99.4% confidence) Before: edm (94.9% confidence) Middle: pop (86.9% confidence)
John Denver — Take Me Home Country Roads	Country	Before: rock (77.3% confidence) Middle: country (52.5% confidence) After: country (50.2% confidence)
One Direction — 18	Pop	After: edm (97.9% confidence) Before: edm (46.0% confidence) Middle: pop (29.9% confidence)
Knox — Girl On The Internet	Electronic Rock	Middle: edm (87.1% confidence) After: edm (65.1% confidence) Before: pop (42.4% confidence)
ILLENIUM — Crawl Outta Love	EDM	Before: edm (98.7% confidence) After: edm (85.7% confidence) Middle: edm (60.9% confidence)
The Beatles — Let It Be	Rock	Middle: rock (81.0% confidence) Before: edm (71.9% confidence) After: classical (56.3% confidence)
Fly Me To The Moon	Jazz	Middle: hiphop (45.3% confidence) After: pop (27.4% confidence) Before: classical (23.4% confidence)

Interpretacja wyników

Wyzwaniem staje się klasyfikacja utworów będących mieszankami kilku gatunków. Coraz więcej utworów Pop lub Rock wykazuje cechy muzyki elektronicznej. Poprawa wyników wymagałaby dziesiątki tysięcy nowoczesnych utworów, których tagowanie samo w sobie okazałoby się kwestią sporną.

W środowisku klasyfikacji do celów pomocy producentom w podejmowaniu decyzji odnośnie korekcji dźwięku, błędna klasyfikacja wcale nie musi oznaczać błędnego aplikowania efektów – techniki mieszają się z biegiem lat. Należy również pamiętać, że żaden program nie osiągnie 100% poprawności oceniając sztukę, która często „łamie zasady" konwencji i tworzy nowe nurty budowane na historycznych zwyczajach.

Podsumowanie

Dodanie do danych treningowych autorskich utworów z gatunku EDM okazało się ciekawym dodatkiem poprawiającym rzeczywistą skuteczność działania klasyfikatora, natomiast (z uwagi na brak reprezentacji utworów z ostatnich lat) nieuczciwie zaburzyło to klasyfikację nowszych utworów z elementami elektronicznymi nienależącymi do kategorii „EDM".

Niemniej jednak nie oznacza to nieadekwatności modelu – rozpoznaje on poprawnie ogólny typ pod względem technicznym, który może napędzić narzędzia podobne do w/w produktów firmy Sonible, lub innych rozpoznawalnych twórców oprogramowania dla producentów muzycznych wspartego uczeniem maszynowym jak Izotope (https://www.izotope.com).

Użyte materiały

GTZAN dataset (https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification)

Autorskie utwory (https://soundcloud.com/wojtek-987):

- All My Fault (remix)
- Hope You Find Some (instrumental)
- Make Me
- Risk – Gracie Abrams (remix)
- Zabierz Mnie (https://open.spotify.com/track/1gLdG7O1oyO4efDl66wJ8G?si=9a94387402e542f2)
- Zedd – Funny (remix)
- Zedd – Funny (remix) (instrumental)

Tensorflow (python) (https://www.tensorflow.org)