Detectarea si transcrierea notelor de chitara bas din melodii

Tiberiu-Andrei Popescu
Universitatea din București
tiberiu-andrei.popescu@s.unibuc.ro

Vlăduț-Ioan Voicu
Universitatea din București
vladut-ioan.voicu@s.unibuc.ro

Teodora-Laura Cojocaru
Universitatea din București
teodora-laura.cojocaru@s.unibuc.ro

***Abstract*—Această lucrare își propune să verifice dacă rezultatele obținute de Abeßer și Schuller în transcrierea chitarei bass pot fi reproduse fidel. De fapt, problema pe care o tratăm este în același timp detectarea notelor, cât și estimarea modului în care instrumentistul a interpretat partitura. Pentru a atinge acest scop, am implementat o arhitectură care împarte problema în două etape. În prima parte, ne concentrăm pe semnalul audio pentru a extrage parametrii de bază, folosindu-ne de un șablon armonic ajustat cu un factor de inarmonicitate pentru a urmări frecvența f_0. Mai departe, luăm acești parametri și îi trecem printr-un clasificator SVM, care are rolul de a ne indica stilul de ciupire și poziția de pe grif.La final, comparăm direct rezultatele implementării noastre cu cele din lucrarea originală, folosind exact același set de date, pentru a evalua măsura în care metoda este într-adevăr reproductibilă.**

***Termeni de Indexare*—Transcriere bass, chitară bass electrică, poziție pe grif, estimare stil de redare, detectarea corzii.**

I. Introducere

Transcrierea Automată a Muzicii rămâne o problemă de actualitate atât în lumea muzicală cât și în lumea procesării de semnale. Diversi algoritmi au fost propuși de-a lungul timpului pentru a transcrie sunete și melodii produse de instrumente muzicale diverse precum tobe, chitară bas sau chiar transcriere polifonică. În toți acești algoritmi, procesul de transcriere constă în identificarea cu acuratețe a tuturor notelor muzicale redate într-un semnal audio. În acest context, o notă muzicală este descrisă de parametrii la nivel de partitură, cum sunt debutul, durata și înălțimea.

Această lucrare se concentrează pe domeniul specific al transcrierii înregistrărilor monofonice de chitară bas. Acesta este un instrument de bază în muzica modernă, însă multitudinea de stiluri de interpretare fac transcrierea melodiilor de chitară bas să fie o provocare.

Acestea fiind spuse, se remarcă lucrarea [1] a lui Abeßer și Schuller, lucrare pe care ținem să o reproducem. Aceștia introduc parametri legați de instrument, pe lângă parametrii la nivel de partitură. Mai exact, se introduc stilul de ciupire al corzilor precum

și stilul de expresie, adică tehnica de redare aplicată corzii după ciupire. În Tabelul I sunt enumerate toate stilurile de ciupire și de expresie considerate în lucrare. De asemenea, autorii introduc și poziția de pe grif ca parametru, reprezentată de numărul corzii și de numărul de rând al grifului (tasta). Pentru a afla acești parametri, aceștia folosesc Support Vector Machines (SVM).

Contribuția științifică pe care o aducem prin această lucrare este verificarea reproductibilității rezultatelor tezei antemenționate, o calitate importantă în lumea învățării automate. Ne propunem să realizăm o implementare independentă a algoritmului descris de Abeßer și Schuller și să comparăm rezultatele obținute cu acelea prezentate de aceștia. De asemenea, vom folosi același set de date pe care autorii lucrării originale l-au folosit, pentru a elimina cât mai multe variabile ce pot influența rezultatele finale.

În continuare, lucrarea este structurată după cum urmează. În secțiunea următoare (II) vom enumera și descrie pe scurt câteva soluții similare care există în climatul academic actual în legătură cu acest domeniu. În secțiunea III vom vorbi în ansamblu despre algoritmul implementat. Apoi, în secțiunea IV vom detalia mai mult despre acesta. În secțiunea V vom prezenta și analiza rezultatele pe care le-am obținut, comparându-le cu lucrarea originală, urmând, în final, secțiunea VI, unde vom rezuma pe scurt descoperirile făcute pe parcursul lucrării.

TABLE I

TAXONOMIA TEHNICILOR DE INTERPRETARE LA CHITARA BAS

Stil de ciupire	**Abr.**	**Stil de expresie**	**Abr.**
Ciupire cu degetele (Finger-Style)	FS	Normal	NO
Ciupire cu pana (Picked)	PK	Armonice (Harmonics)	HA
Atenuat (Muted)	MU	Note moarte (Dead-Notes)	DN
Slap (deget mare)	ST	Vibrato	VI
Slap (ciupire/pluck)	SP	Bending	BE
		Slide	SL

II. LUCRĂRI RELEVANTE

A. Score Parameter Estimation

Transcrierea automată a muzicii este, în general, formulată ca o problemă de estimare a parametrilor de tip *score* (înălțime, moment de început, durată) pornind direct de la semnalul audio, fără a impune inițial constrângeri specifice unui instrument anume. O trecere în revistă a principalelor direcții și dificultăți este oferită în [2], unde sunt discutate atât metode clasice bazate pe analiză spectrală, cât și abordări moderne bazate pe învățare automată.

Pentru chitara bas, multe lucrări pornesc de la ipoteza că linia de bas este cea mai joasă voce monofonică din mix, ceea ce permite aplicarea unor strategii dedicate de transcriere. De exemplu, [3] propune una dintre primele metode de transcriere automată a liniei de bas din muzică polifonică, iar lucrări ulterioare au extins această idee prin integrarea de modele acustice și muzicologice [4]. Alte sisteme folosesc baze de date de trăsături audio pentru estimarea simultană a melodiei și a liniei de bas [5] sau analizează tipare de linie de bas pentru sarcini precum clasificarea de gen muzical [6].

În ceea ce privește reprezentarea spectrală, multe

instrumente software de transcriere polifonică se bazează pe STFT și pe variante ale acesteia [7]. Totuși, limitările de rezoluție în frecvențele joase au condus la explorarea unor tehnici mai avansate de estimare a frecvenței fundamentale și de urmărire a componentelor armonice. Sistemele de descriere a scenei muzicale în timp real, precum cel propus în [8], combină estimarea predominantă a lui f_0 cu detectarea liniei melodice și a liniei de bas în semnale reale, iar lucrări ulterioare au arătat că integrarea explicită a informației de percuție și a liniilor de bas poate îmbunătăți semnificativ clasificarea muzicală [9].

Estimarea frecvenței fundamentale este adesea realizată fie prin construirea unei funcții de salientă – o funcție care indică, pentru fiecare frecvență posibilă, cât de probabil este ca aceasta să corespundă înălțimii percepute –, fie prin urmărirea în timp a contururilor de înălțime De exemplu, [10] propune o funcție de salientă bazată pe cromă, o reprezentare care grupează energia spectrală în cele 12 clase de înălțime muzicală (C, D etc.), indiferent de octavă, facilitând astfel extragerea melodiei și a liniei de bas. In schumb, [11] introduce pYIN, un estimator de f_0 care folosește distribuții probabilistice de prag pentru a obține contururi de înălțime robuste în contexte monofonice și ușor polifonice.

În contextul acestui proiect, ne-am aliniat conceptual cu abordările centrate pe bas din [1], [12], [13], dar am adaptat pipeline-ul la scenariul nostru de înregistrări solo: am păstrat ideea de reprezentare spectrală bogată în informație (inspirată de [7], [8]) și am rafinat etapele de preprocesare și detecție de onset pentru a obține estimări mai stabile ale parametrilor de tip score.

B. Instrument Parameter Estimation

Dincolo de parametrii de tip *score*, o direcție importantă este transcrierea centrată pe instrument, în care se încearcă estimarea poziției pe grif, a corzii și a tehnicilor de interpretare. Pentru vioară, de exemplu, [14] propune descriptorii de nivel jos pentru transcriere automată, iar lucrări ulterioare au dezvoltat sisteme de urmărire în timp real a poziției arcușului și a gesturilor instrumentistului [15], [16]. Analize similare pentru instrumente cu coarde frecate, precum violoncelul, sunt prezentate în [17].

Pentru chitară și bas, literatura acoperă atât analiza exclusiv audio, cât și abordări multimodale. Pe partea audio, [18] propune un sistem de transcriere în timp real pentru chitară electrică, iar [19] extinde ideea către generarea automată de tablaturi prin estimarea simultană a parametrilor de tip score și a celor specifici instrumentului. Estimarea multipitch și modelarea probabilistă a spectrului pentru pian sunt discutate în [20], oferind un cadru general pentru tratarea sunetelor polifonice cu structură armonică bogată.

Estimarea tehnicilor de interpretare și a parametrilor expresivi a fost abordată prin modele statistice și rețele neuronale. De exemplu, [21] modelează controlul arcușului pentru sinteza sunetului de vioară, iar [22] propune un model de

timbru bazat pe rețele neuronale pentru controlul performanței la vioară. Pentru chitară clasică, [23] extrage parametri fizici și expresivi pentru sinteză model-based, în timp ce lucrări precum [24]–[26] combină senzori de mișcare sau senzori capacitivi cu analiza audio pentru a detecta tehnici precum hammer-on, pull-off, slide sau variații de atac.

Estimarea poziției pe grif și a corzii a fost studiată atât pentru chitară, cât și pentru bas. [27] abordează problema extragerii poziției de ciupire și a degetării pe coardă dintr-o simplă înregistrare audio, iar [28] extinde ideea către detectarea expresivității în muzica pentru vioară. În zona chitarei electrice, [29] folosește modele bazate pe secvențe de f_0 pentru a detecta tehnici de interpretare în înregistrări reale.

În raport cu aceste lucrări, proiectul nostru se aliniază conceptual cu abordarea centrată pe instrument din [1], [12], [13], dar o specializează pentru scenariul de înregistrări solo de chitară bas. Am păstrat ideea de a combina parametri de tip score cu parametri specifici instrumentului (corzi, poziții pe grif, indicii timbrali), însă am rafinat etapele de detecție a onset-urilor și de preprocesare pentru a obține un pipeline mai robust și mai ușor de integrat într-un sistem de analiză și vizualizare a performanței.

III. Arhitectura

Pentru a ne atinge scopul de a transcrie melodii de chitară bas, propunem o arhitectură împărțită în două etape principale (fig. 1), după cum sunt descrise în continuare.

Pentru a ne atinge scopul de a transcrie melodii de chitară bas, propunem o arhitectură împărțită în două etape principale (fig. 1), după cum sunt descrise în continuare.

Prima etapă presupune extragerea de caracteristici din datele de antrenare. Această etapă constă în pre-procesare și estimare spectrală, detectarea debutului notelor, detectarea și urmărirea parcursului frecvenței fundamentale f_0, detectarea sfârșitului notei, modelarea învelișului spectral și, în final, estimarea intensității și segmentarea notelor în atac și decădere. Caracteristicile extrase vor fi păstrate ca parametri la nivel de partitură și vor fi date mai departe ca intrare algoritmului de clasificare Support Vector Machine (SVM), care va estima tehnica de redare a sunetelor și pozițiile pe grif necesare producerii lor.

Aceasta va fi etapa a doua, ce constă în clasificare folosind SVM și calcularea poziției pe grif. La finalul acestui proces, vom rămâne cu parametrii de scor extrași în prima etapă și parametrii de instrument extrași din etapa a doua. Aceștia ne vor da o imagine de ansamblu completă, având o transcriere fidelă a partiturii, cât și o estimare a modului și stilului de interpretare.

IV. Implementare

A. Pre-procesare și Estimare Spectrală

1) Downsampling: În prima etapă a proiectului am implementat un modul de downsampling al

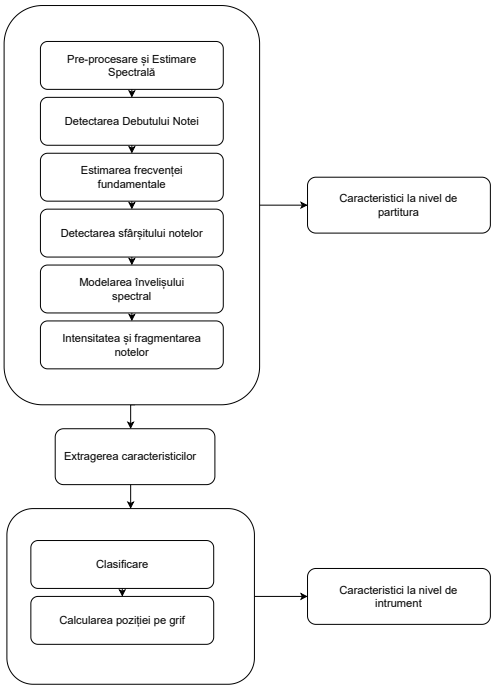

Fig. 1. Diagrama arhitecturii alogritmului implementat.

semnalului audio, care reduce rata de eșantionare de la valoarea originală la o nouă valoare țintă.

$$f_{s,target} = 5512.5 \text{ Hz} \tag{1}$$

În prima versiune a proiectului, downsampling-ul semnalului audio a fost realizat prin interpolare liniară, reconstruind semnalul la noile poziții temporale corespunzătoare ratei țintă. Această metodă are avantajul simplității și păstrează continuitatea formei de undă, evitând artefactele severe care apar în cazul unei decimări brute (luarea fiecărui al k-lea eșantion). Totuși, interpolarea liniară nu include niciun mecanism de anti-aliasing, ceea ce înseamnă că toate componentele spectrale aflate peste noua frecvență Nyquist sunt pliate în banda utilă. În cazul chitarei bas, unde pot exista armonici și zgomote peste 3–4 kHz, acest lucru introduce aliasing vizibil și degradează analiza ulterioară.

Pentru a elimina aceste probleme, am înlocuit complet abordarea inițială. În noua implementare, semnalul este mai întâi trecut printr-un filtru trece-jos Butterworth de ordinul 8, proiectat astfel încât să atenueze toate componentele spectrale aflate peste banda utilă înainte de reducerea ratei de eșantionare. Filtrul Butterworth este o alegere standard deoarece are un răspuns în frecvență foarte neted, fără ondulații în bandă (*ripple*), și o tranziție relativ abruptă între banda trece și banda oprește. Acest comportament îl face potrivit pentru anti-aliasing: frecvențele aflate peste limita impusă de noul Nyquist sunt suprimate progresiv, astfel încât, după downsampling, ele nu se mai pot replia în banda utilă. În implementarea noastră, frecvența de tăiere este setată la

$$0.9 \cdot \frac{f_{s,\text{target}}}{2},$$

o valoare puțin sub limita teoretică, pentru a evita efectele de margine și pentru a asigura o atenuare suficientă înainte de tranziția către banda oprește.

După filtrare, resampling-ul propriu-zis este realizat cu funcția `scipy.signal.resample_poly`. Această metodă exprimă raportul dintre rata originală și rata țintă sub forma unui raport rațional simplificat

$$\frac{\text{up}}{\text{down}},$$

unde *up* reprezintă factorul de supraeșantionare, iar *down* factorul de decimare. În practică, semnalul este mai întâi supraeșantionat cu factorul *up*, apoi filtrat cu un filtru FIR intern de calitate ridicată, iar în final decimat prin păstrarea fiecărui al *down*-lea eșantion. Această structură polifazică permite o conversie foarte precisă între rate arbitrare, evitând artefactele introduse de metodele naive de interpolare. În plus, filtrul Butterworth aplicat anterior garantează că semnalul nu conține energie peste noul Nyquist, astfel încât verificarea manuală a teoremei Nyquist nu mai este necesară: aliasing-ul este prevenit direct prin filtrare.

Noua versiune a modulului de downsampling produce un semnal curat, fără artefacte de aliasing, cu o structură spectrală stabilă și perfect adecvată pentru etapele ulterioare ale pipeline-ului. Această îmbunătățire a crescut semnificativ calitatea analizei, reducând în același timp costul computațional datorită ratei de eșantionare mai mici.

2) STFT și estimarea spectrului: În etapa următoare am implementat calculul transformatei Fourier pe ferestre scurte (STFT), care reprezintă fundamentul tuturor pașilor ulteriori.

Pentru fiecare cadru al semnalului aplicăm o fereastră Hann, apoi calculăm FFT-ul zero-padding-uit la 4096 de puncte, obținând o rezoluție spectrală foarte fină în zona frecvențelor joase, unde se află informația relevantă pentru chitara bas. De asemenea, am folosit un hopsize de 32 de eșantioane.

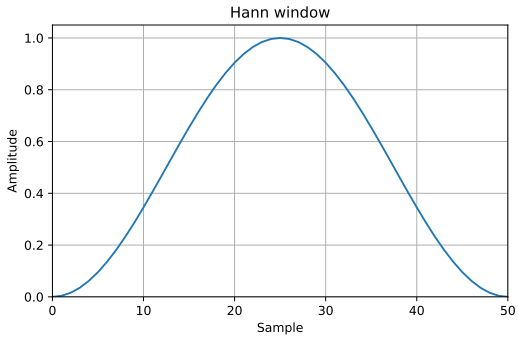

Fig. 2. Hann window

Am folosit o fereastră Hann deoarece aceasta reduce semnificativ scurgerile spectrale (”spectral leakage”), o problemă frecventă atunci când analizăm semnale armonice cu ferestre finite. Formula ferestrei este:

$$w[n] = 0.5\left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right) \qquad (2)$$

Această fereastră atenuează marginile cadrului, minimizează interferența dintre armonici și produce un spectru mai curat și mai stabil. Pentru instrumente cu armonici bine definite, cum este basul, reducerea scurgerilor spectrale este esențială pentru a putea urmări corect frecvența fundamentală și armonicile.

Pentru fiecare cadru de 512 eșantioane aplicăm zero-padding până la 4096 de puncte înainte de FFT. Zero-padding-ul nu adaugă informație nouă, dar: crește densitatea punctelor în domeniul frecvenței, permite localizarea mai precisă a vârfurilor armon-

ice, îmbunătățește estimarea frecvenței instantanee, produce o spectrogramă mai fină, necesară pentru kernelul de onset detection.

În contextul chitarei bas, unde diferențele dintre tehnici (slap, fingerstyle, muted) se reflectă în structura armonicilor, această rezoluție suplimentară este foarte valoroasă.

Axa de frecvență a spectrogramelor STFT este liniară și se calculează folosind formula:

$$f(k) = \frac{k}{N_{FFT}} \cdot f_s \tag{3}$$

În această expresie:

1) k este indexul bin-ului FFT.
2) N_{FFT} este lungimea FFT-ului (în cazul nostru 4096), adică numărul total de puncte în care este eșantionat spectrul.
3) f_s rata de eșantionare a semnalului (5512.5 Hz după downsampling).
4) Termenul următor reprezintă fracția din banda totală de frecvență ocupată de bin-ul k:

$$\frac{k}{N_{FFT}} \tag{4}$$

Înmulțind această fracție cu f_s, obținem frecvența reală (în Hz) corespunzătoare acelui bin. Intervalul următor reflectă faptul că folosim FFT reală (rfft), care produce doar jumătate din spectru (partea pozitivă), până la frecvența Nyquist:

$$k \in [0, N_{FFT}/2] \tag{5}$$

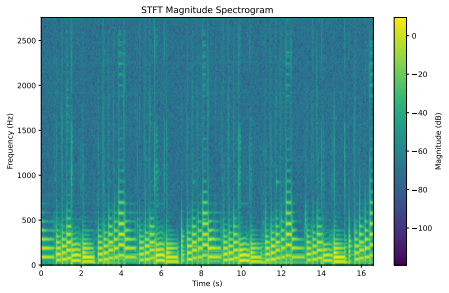

Fig. 3. Spectograma magnitudinilor din STFT

3) Instantaneous Frequency Estimation & Reassigned Log-Frequency Spectrogram (MIF): În această etapă am implementat estimarea frecvenței instantanee (Instantaneous Frequency, IF) pentru fiecare bin timp-frecvență al STFT-ului, urmată de construirea spectrogramei reasignate în frecvență logaritmică (MIF), exact după metodologia descrisă de Lagrange & Marchand și utilizată în [1].

Pentru a reprezenta armonicile și variațiile fine ale frecvenței fundamentale într-un mod muzical relevant, folosim o axă de frecvență logaritmică cu 10 bin-uri pe semiton. Această rezoluție ridicată este necesară pentru a surprinde modulațiile mici ale lui f_0 caracteristice tehnicilor expresive precum vibrato, bending și slide.

Axa este definită prin:

$$f_{log}(k_{log}) = 440 \cdot 2^{\frac{22+k_{log}/10-69}{12}} \tag{6}$$

Unde $k_{log} \in [0, 780]$. În această formulă:

- k_{log} este indexul pe axa logaritmică (0-780), corespunzând unui interval de 78 semitonuri.

- 10 bin-uri/semiton $\rightarrow$ rezoluție foarte fină pentru urmărirea lui f_0. $f_{ref} = 440$ Hz este nota A4.
- 22-69 ajustează scala astfel încât primul bin să fie în jur de 29.1 Hz (A#0), iar ultimul să ajungă la $f_s/2 \approx 2621.8$ Hz, acoperind întreaga bandă utilă pentru chitara bas.

Această axă transformă coordonata verticală a spectrogramei dintr-un simplu index FFT într-o scară muzicală reală.

Pentru a aplica metoda de reassignment a lui Lagrange & Marchand, avem nevoie de două STFT-uri: STFT cu fereastra Hann ($w[n]$) și STFT cu derivata ferestrei ($w'[n]$). Derivata analitică a ferestrei Hann este:

$$w'(n) = \frac{\pi}{L-1} \sin\left(\frac{2\pi n}{L-1}\right) \tag{7}$$

Unde:

- L este lungimea ferestrei (512).
- Derivata este necesară pentru a estima derivata temporală a fazei locale.
- Aceasta intră direct în formula de reassignment (Eq. 17-19 în Lagrange & Marchand [30]).

Pentru estimarea frecvenței instantanee (IF) folosim metoda de reassignment bazată pe fază:

$$\hat{f}(k,n) = f_{bin}(k) - \frac{1}{2\pi}\text{Im}\left(\frac{S_{w'}(k,n)}{S_w(k,n)}\right) \tag{8}$$

Unde:

- $S_w(k,n)$ este STFT-ul cu fereastra Hann.
- $S_{w'}(k,n)$ este STFT-ul cu derivata ferestrei.
- Raportul lor estimează derivata temporală a fazei.
- Termenul de corecție ajustează frecvența din centrul bin-ului FFT către frecvența reală a sinusoidei.

Pentru stabilitate numerică, folosim un mask care ignoră bin-urile cu energie foarte mică:

$$|S_w(k,n)|^2 > 10^{-10} \max|S_w|^2 \tag{9}$$

Demonstrația formulei de reassignment: În articolul lui Lagrange & Marchand [30], punctul de plecare este STFT-ul unui semnal fereastrat:

$$S_w(\omega,t) = \int s(\tau)w(\tau-t)e^{-j2\pi\omega(\tau-t)}d\tau \tag{10}$$

Acest spectru este complex și poate fi scris în formă polară:

$$S_w(\omega,t) = A(\omega,t)e^{j\phi(\omega,t)} \tag{11}$$

Unde $A(\omega,t)$ este amplitudinea, iar $\phi(\omega,t)$ este faza locală. Dacă aplicăm logaritmul acestei formule obținem:

$$\log S_w(\omega,t) = \log A(\omega,t) + j\phi(\omega,t) \tag{12}$$

Ideea de bază este că frecvența instantanee într-un punct timp-frecvență este dată de derivata temporală a fazei:

$$\omega_{inst}(\omega,t) = \frac{1}{2\pi}\frac{\partial\phi(\omega,t)}{\partial t} \tag{13}$$

În loc să derive direct faza, autorii derivă logaritmul lui S_w:

$$\frac{\partial}{\partial t}\log S_w(\omega,t) = \frac{\partial}{\partial t}\log A(\omega,t) + j\frac{\partial \phi(\omega,t)}{\partial t} \quad (14)$$

Partea imaginară a acestei derivate este exact derivata fazei:

$$\text{Im}\left(\frac{\partial}{\partial t}\log S_w(\omega,t)\right) = \frac{\partial \phi(\omega,t)}{\partial t} \quad (15)$$

Asta este exact Eq. (14) din [30], scris explicit:

$$\omega_{inst}(\omega,t) = \frac{1}{2\pi}\frac{\partial \phi(\omega,t)}{\partial t} = \frac{1}{2\pi}\text{Im}\left(\frac{\partial}{\partial t}\log S_w(\omega,t)\right) \quad (16)$$

Folosind formula de derivare a logaritmului:

$$\frac{\partial}{\partial t}\log S_w(\omega,t) = \frac{1}{S_w(\omega,t)}\frac{\partial}{\partial t}S_w(\omega,t) \quad (17)$$

Obținem:

$$\omega_{inst}(\omega,t) = \frac{1}{2\pi}\text{Im}\left(\frac{1}{S_w(\omega,t)}\frac{\partial}{\partial t}S_w(\omega,t)\right) \quad (18)$$

Calculăm separat $\frac{\partial}{\partial t}S_w(\omega,t)$:

$$S_w(\omega,t) = \int_{-\infty}^{+\infty} s(\tau)w(\tau-t)e^{-j2\pi\omega(\tau-t)}d\tau \quad (19)$$

Derivata în timp intră în integrală (s este funcție de τ, nu de t):

$$\frac{\partial}{\partial t}S_w(\omega,t) = \int_{-\infty}^{+\infty} s(\tau)\frac{\partial}{\partial t}\left[w(\tau-t)e^{-j2\pi\omega(\tau-t)}\right]d\tau \quad (20)$$

Aplicăm regula produsului:

$$\frac{\partial}{\partial t}[w(\tau-t)e^{-j2\pi\omega(\tau-t)}] = \frac{\partial w(\tau-t)}{\partial t}e^{-j2\pi\omega(\tau-t)} + w(\tau-t)\frac{\partial}{\partial t}e^{-j2\pi\omega(\tau-t)} \quad (21)$$

Acum derivăm fiecare termen. Deci:

- $\frac{\partial}{\partial t}w(\tau-t) = -w'(\tau-t)$ (derivata în timp a ferestrei deplasate).
- $\frac{\partial}{\partial t}e^{-j2\pi\omega(\tau-t)} = j2\pi\omega\ e^{-j2\pi\omega(\tau-t)}$ (pentru că derivata lui $-(\tau-t)$ față de t este $+1$).

Deci:

$$\frac{\partial}{\partial t}[w(\tau-t)e^{-j2\pi\omega(\tau-t)}] = -w'(\tau-t)e^{-j2\pi\omega(\tau-t)} + j2\pi\omega\ w(\tau-t)e^{-j2\pi\omega(\tau-t)} \quad (22)$$

Introducem asta în derivata lui S_w:

$$\frac{\partial}{\partial t}S_w(\omega,t) = \int s(\tau)[-w'(\tau-t)e^{-j2\pi\omega(\tau-t)} + j2\pi\omega\ w(\tau-t)e^{-j2\pi\omega(\tau-t)}]d\tau \quad (23)$$

Separăm cei doi termeni:

$$\frac{\partial}{\partial t}S_w(\omega,t) = -\int s(\tau)w'(\tau-t)e^{-j2\pi\omega(\tau-t)}d\tau + j2\pi\omega\int s(\tau)w(\tau-t)e^{-j2\pi\omega(\tau-t)}d\tau \quad (24)$$

Observăm acum că:

- Al doilea integral este chiar $S_w(\omega,t)$.

- Primul integral este STFT-ul calculat cu derivata ferestrei, notat $S_{w'}(\omega, t)$.

Deci:

$$\frac{\partial}{\partial t} S_w(\omega, t) = -S_{w'}(\omega, t) + j2\pi\omega \; S_w(\omega, t) \quad (25)$$

Acum înlocuim în formula pentru $\omega_{inst}(\omega, t)$:

$$\omega_{inst}(\omega, t) = \frac{1}{2\pi} \text{Im} \left(\frac{-S_{w'}(\omega, t) + j2\pi\omega \; S_w(\omega, t)}{S_w(\omega, t)} \right) \quad (26)$$

Simplificând:

$$\omega_{inst}(\omega, t) = \frac{1}{2\pi} \text{Im} \left(j2\pi\omega - \frac{S_{w'}(\omega, t)}{S_w(\omega, t)} \right) \quad (27)$$

Aici:

- $\text{Im}(j2\pi\omega) = 2\pi\omega$.
- $\text{Im} \left(-\frac{S_{w'}(\omega,t)}{S_w(\omega,t)} \right) = -\text{Im} \left(\frac{S_{w'}(\omega,t)}{S_w(\omega,t)} \right)$.

Deci formula finală este:

$$\omega_{inst}(\omega, t) = \omega - \frac{1}{2\pi} \text{Im} \left(\frac{S_{w'}(\omega, t)}{S_w(\omega, t)} \right) \quad (28)$$

Unde:

- ω este frecvența centrală a bin-ului FFT (frecvența ”grosieră”).
- Termenul $-\frac{1}{2\pi} \text{Im} \left(\frac{S_{w'}(\omega,t)}{S_w(\omega,t)} \right)$ este termenul de corecție care deplasează frecvența din centrul bin-ului către frecvența reală a sinusoidei.

În implementarea discretă, ω devine frecvența centrală a bin-ului FFT:

$$f_{bin}(k) = \frac{k}{N_{FFT}} f_s \quad (29)$$

După ce avem frecvența instantanee pentru fiecare bin (k, n), reasignăm energia către axa logaritmică și construim spectrograma reassigned MIF:

1) Pentru fiecare bin (k, n) din STFT:
   - Calculăm frecvența instantanee $\hat{f}(k, n)$ în Hz.
   - Găsim bin-ul logaritmic k_{log} cu frecvența cea mai apropiată.
   - Adăugăm magnitudinea $(M(k, n))$ în acel bin.
2) Pentru fiecare timp n, acumulăm toate valorile care se reasignează în același bin logaritmic.

Rezultatul este matricea $M_{IF} \in \mathbb{R}^{K_{log} \times N}$, unde $K_{log} = 780$ (axa logaritmică) și N este numărul de cadre.

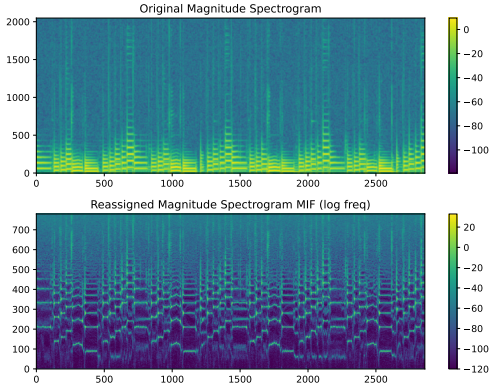

Fig. 4. Spectograma initiala si spectograma rearignata (MIF)

O spectrogramă reasignată în frecvență logaritmică (MIF) oferă o reprezentare mult mai precisă și mai expresivă a semnalelor armonice decât spectrograma STFT clasică, deoarece redistribuie energia fiecărui bin către frecvența instantanee estimată,

eliminând astfel efectul de ”împrăștiere” caracteristic analizei pe grilă liniară. În locul petelor late din STFT, armonicile apar ca linii subțiri și bine definite, ceea ce permite urmărirea clară a variațiilor fine ale frecvenței fundamentale, inclusiv tehnici expresive precum vibrato, bending sau slide.

Această concentrare a energiei în jurul frecvențelor reale îmbunătățește semnificativ robustețea estimării lui f_0, deoarece fundamentalul și armonicile sale devin vârfuri distincte, nu regiuni difuze. În plus, atacurile instrumentale sunt mult mai evidente în MIF, ceea ce duce la o detecție a onset-urilor mai precisă și mai stabilă. În ansamblu, MIF oferă o reprezentare timp-frecvență mult mai fidelă comportamentului fizic al semnalului, fiind astfel superioară STFT-ului clasic în toate etapele de analiză spectrală relevante pentru transcrierea chitarei bas.

B. Detectarea Debutului Notei

Pentru detecția începuturilor de notă (note onsets), am implementat pipeline-ul propus în [1], bazat pe măsurarea ”noutății armonice” în spectrograma de frecvență instantanee MIF. Ideea centrală este că un atac instrumental produce o schimbare bruscă în structura armonicilor, iar această schimbare poate fi detectată printr-un filtru bidimensional special conceput.

Kernelul folosit este construit ca produsul a doi vectori:

$$K = \begin{bmatrix} 0.3 \\ 1.0 \\ 1.0 \\ 1.0 \\ 0.3 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 0 & -1 & -1 & -1 \end{bmatrix} \quad (30)$$

Acest kernel are două roluri fundamentale:

- Smoothing pe axa de frecvență - agregă energia armonicilor apropiate, ceea ce stabilizează reprezentarea și reduce zgomotul spectral.
- Detecția de muchii pe axa temporală - partea $([1, 1, 1, 0, -1, -1, -1])$ funcționează ca un filtru diferențial care răspunde puternic la tranzițiile bruște specifice atacurilor instrumentale.

Aplicăm convoluția bidimensională: folosind convolve2d cu padding zero:

$$M_{IF,K} = M_K *_{2D} \text{MIF} \quad (31)$$

Rezultatul este o hartă timp-frecvență în care atacurile sunt evidențiate prin valori mari, deoarece kernelul este un filtru ”matched” pentru structura tipică a unui onset.

Pentru fiecare cadru temporal (n), extragem maximul pe axa de frecvență:

$$\alpha_{on}(n) = \max_k M_{IF,K}(k, n) \quad (32)$$

Această operație comprimă informația spectrală într-un singur scalar per cadru, obținând o curbă

temporală care prezintă vârfuri clare în dreptul atacurilor instrumentale.

Folosim un prag fix de 20% din maximul global și selectăm toate maximele locale care depășesc acest prag.

Mai departe, găsim începuturile de notă folosind funcția `find_peaks`. Funcția `find_peaks` implementează un algoritm de detecție a maximelor locale într-un semnal unidimensional. Un peak este identificat atunci când valoarea curentă este mai mare decât vecinul din stânga și cel puțin egală cu vecinul din dreapta.

Funcția permite două criterii suplimentare: un prag minim de amplitudine (height) și o distanță minimă între vârfuri (distance). Dacă două peak-uri apar la o distanță mai mică decât cea impusă, algoritmul păstrează doar peak-ul cu amplitudinea mai mare, eliminând astfel detecțiile redundante cauzate de fluctuații locale ale semnalului. Rezultatul final este lista indecșilor cadrelor în care apar vârfuri semnificative, utilă în special pentru detecția onset-urilor.

Am implementat și o funcție proprie `find_peaks`, dar în testele noastre am obținut rezultate mai stabile folosind versiunea optimizată din `scipy.signal.find_peaks`, motiv pentru care am păstrat-o în varianta finală a pipeline-ului.

Pe lângă pipeline-ul descris în [1], am identificat în practică mai multe situații în care metoda originală producea detecții eronate. În special, am observat o supra-detectare semnificativă atunci când

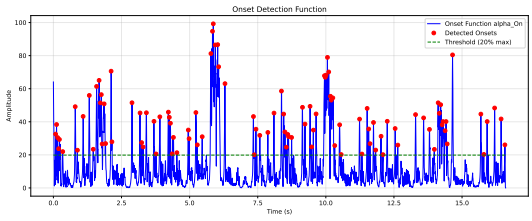

Fig. 5. Detectarea debutului notelor

pragul era setat la valoarea recomandată de 0.20. Ajustând pragul la 0.30, numărul de detecții false a scăzut considerabil. De asemenea, am mărit distanța minimă dintre două onset-uri consecutive de la 5 la 15 cadre (aprox. 87 ms), ceea ce reduce detecțiile multiple în interiorul aceluiași atac instrumental. În plus, am identificat un edge-case care nu este tratat în articol: kernelul diferențial nu poate detecta un onset aflat chiar la începutul semnalului, deoarece nu există suficiente cadre anterioare pentru a forma o muchie temporală. Pentru a remedia această situație, am introdus o verificare suplimentară care detectează un onset la cadrul $n = 0$ dacă energia din primele cadre depășește un prag relativ. Această completare asigură detecția corectă a debuturilor de notă chiar și în cazul în care prima notă începe imediat la $t = 0$.

C. *Estimarea frecvenței fundamentale*

După detectarea debutului notelor, estimăm frecvența fundamentală f_0. Acest pas presupune medierea lui M_{IF} peste primele 20% intervale de frecvență dintre două debuturi consecutive, obținând $M_{IF,acc}$. Medierea este necesară pentru a stabiliza

semnalul emis în primele momente ale începutului notei analizate. Mai departe, definim un șablon $T(f, \beta)$ care va avea vârfurile la:

$$f_h = (h+1) f_0 \sqrt{1 + \beta(h+1)^2}, \quad h \in \{0, \ldots, 9\} \tag{33}$$

reprezentând frecvențele armonice ale lui f. De fapt, acest șablon arată foarte similar cu un pieptene Dirac finit. β este un factor de inarmonicitate, care ”întinde” pieptenele spre final (fig. 6). Acest factor este necesar deoarece corzile unei chitare bass nu sunt ideale, astfel mișcând frecvențele armonicelor. Pentru o acuratețe mai mare, primele două magnitudini ale pieptenelui sunt duble față de restul, pentru a magnifica frecvențele mai joase de bass.

Mai departe, vom căuta în grilă candidați f_0 și β optimi. Vom căuta 100 de valori ale lui β în intervalul:

$$\beta \in [0, 0.001] \tag{34}$$

și vom defini o frecvență minimă și una maximă între care vom căuta f_0, alegând candidați din intervalele de frecvență ale $M_{IF,acc}$. Apoi, vom calcula frecvențele f_h după formula (33) și vom maximiza suma magnitudinilor acestor frecvențe din $M_{IF,acc}$ (fig. 7, 8), după formula:

$$L(\hat{\beta}) = \sum_{h=0}^{9} \hat{a}_{h,\hat{\beta}} \tag{35}$$

unde magnitudinile respective sunt interpolate liniar din cele două intervale de frecvență consecutive între care f_h se încadrează:

$$\hat{a}_{h,\hat{\beta}} = (1-w) \cdot M_{IF,acc}[k_{floor}] + w \cdot M_{IF,acc}[k_{floor}+1] \tag{36}$$

unde

$$w = k^* - \lfloor k^* \rfloor \tag{37}$$

iar

$$k^* = 10 \left(12 \log_2 \left(\frac{f_h}{440} \right) + 47 \right) \tag{38}$$

Vom converti apoi frecvența f_0 optimă la MIDI Pitch P:

$$\mathcal{P}(i) = \lfloor 12 \log_2(f_0/440) + 69 + 0.5 \rfloor \tag{39}$$

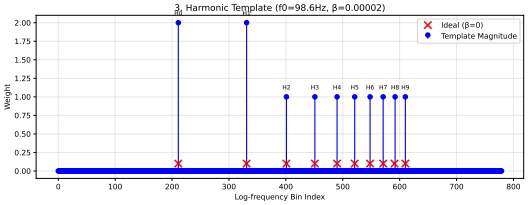

Fig. 6. Șablonul armonic obținut pentru f_0 și β din figură; punctele X roșii reprezintă armonicele ideale.

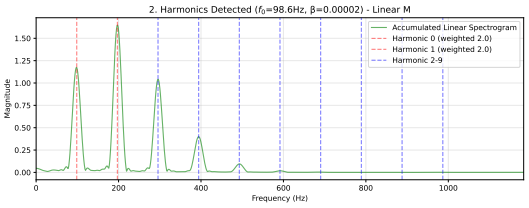

Fig. 7. Spectograma agregată $M_{IF,acc}$ și armonicele detectate.

Mai departe, în al doilea pas, vom urmări evoluția frecvenței fundamentale f_0 găsite (fig. 9). Acest pas este necesar pentru a detecta eventualul sfârșit al notei (proces descris în detaliu în următoarea secțiune). Vom începe de la cadrul n_0, care este

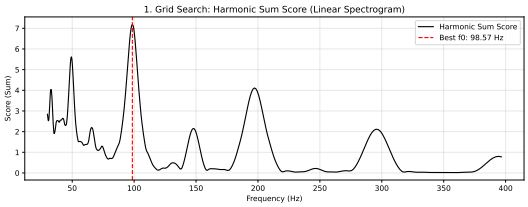

Fig. 8. Scorul obținut de fiecare candidat f_0 (suma amplitudinilor interpolate pentru frecvențele f_h) și candidatul optim.

localizat la distanță de 10% de la debutul notei. Apoi, vom face o trecere înapoi, urmată de o trecere înainte în care vom selecta candidați ai f_0 dintre vecinii intervalului de frecvență al lui f_0 actual din M_{IF}. Pentru a selecta candidatul optim, vom corela încrucișat fiecare cadru al M_{IF} dintre două debuturi consecutive de note cu template-ul armonic spectral translatat. Vom selecta candidatul care obține scorul maxim. Vom stoca conturul $C_{max}(n)$ al acestui scor, pentru a ne ajuta mai departe la calcularea sfârșitului notei (fig. 10).

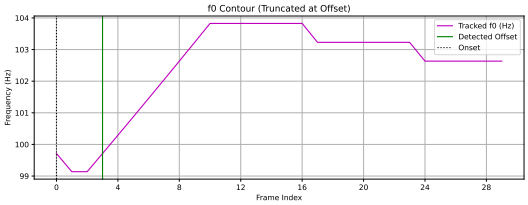

Fig. 9. Conturul (evoluția) lui f_0 în intervalul dintre două note consecutive.

D. Detectarea sfârșitului notelor

Sfârșitul notei este localizat în acel punct din intervalul dintre două note consecutive în care urmează 4 cadre neîntrerupte al căror scor de corelare se află sub o limită reprezentând 5% din scorul maxim de corelare (C_{global_max}) obținut în acest interval, sau atunci când se ajunge la cadrul debutului următoarei note (fig. 10). Vom nota acest cadru găsit cu n_{off}, iar cadrul de început al notei cu n_{on}.

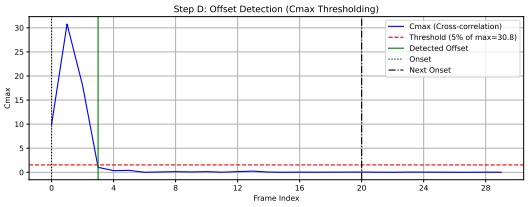

Fig. 10. Conturul (evoluția) lui $C_{\max}(n)$, adică scorul de corelație încrucișată dintre M_{IF} și șablonul spectral.

E. Modelarea înveliș ului spectral

Pentru a modela înveliș ul spectral al notei, vom lua conturul frecvenței fundamentale $f_0(n)$ pe care l-am găsit mai devreme. Apoi, vom calcula frecvențele armonice f_h, după formula (33). Mai departe, vom estima amplitudinile $a_h(n)$ ale acestor armonice găsite interpolând liniar între cele două intervale de frecvență din M_{IF} între care se află fiecare f_h (fig. 11).

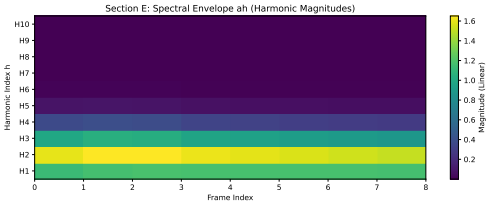

Fig. 11. Înveliș ul spectral $a_h(n)$ al unei note analizate.

F. Intensitatea și fragmentarea notelor

Pentru a clasifica stilul de interpretare, trebuie extrase informații care descriu timbrul și înveliș ul

spectral al notei. Pentru atingerea acestui scop, vom agrega învelișul spectral $a_h(n)$ găsit în secțiunea anterioară, obținând $a(n)$, și vom găsi amplitudinea maximă din intervalul de debut și sfârșit al notei analizate. Astfel, vom găsi cadrul

$$n_{peak} = \underset{n}{\operatorname{argmax}}\, a(n) \tag{40}$$

care va împărți nota în atac (n_{on} și n_{peak}) și decădere (n_{peak} spre n_{off}) (fig. 12). În final, vom calcula intensitatea notei:

$$\mathcal{L}(i) = 20 \log_{10} a(n_{peak}) \tag{41}$$

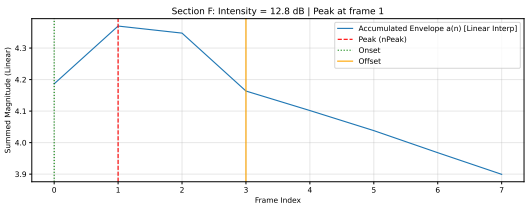

Fig. 12. Segmentarea semnalului notei muzicale în atac și decădere.

G. Extragerea caracteristicilor

După extragerea onset-ului, offset-ului și a frecvenței fundamentale a fiecărei note, următorul pas din implementare este să modelăm natura notei, identificând tehnica de ciupire prin care a fost produsă și diverse tehnici stilistice specifice instrumentului. Pentru a putea observa aceste trăsături ale sunetului, vom analiza notele și mai apoi vom extrage un set de caracteristici care să indice calitățile menționate anterior.

Intuiția este că fiecare tehnică de cântare lasă o amprentă distinctă asupra semnalului. De exemplu, când se folosește tehnica slap (lovirea corzii sau ciupirea acesteia în așa manieră încât să lovească tastele de pe grif), aceasta produce un atac puternic, percutant, urmat de o decădere agresivă. Însă tehnica tradițională, finger-style, produce o notă cu un atac mai calm și o decădere prelungită (dacă aceasta nu este amuțită intenționat, tehnică ce este des întâlnită). Scopul nostru este să captăm aceste diferențe printr-un set de indicatori matematici, care pot fi utilizați de un clasificator în etapele ulterioare. Sunt extrase in paralel, in cazul nostru folosind procesoare AMD generatia a 4-a cu 32 core-uri și generația a 5-a cu 16 core-uri, obținând un timp de aproximativ 17 minute pentru extragerea caracteristicilor (aceasta fiind operația cea mai stresantă computațional, antrenarea decurgând rapid).

În implementarea noastră, extragem aproximativ 212 de astfel de indicatori pentru fiecare notă. Îi putem grupa în mai multe categorii, fiecare captând o caracteristică a sunetului, însă, în general, reutilizăm câteva unelte statistice pentru a augmenta numărul acestora.

1) Evoluția în timp a amplitudinii: Prima categorie descrie cum evoluează volumul în timp al sunetului pe parcursul notei. Orice notă are două faze principale: atacul, creșterea de la liniște la amplitudinea maximă, și decăderea, scăderea de la maxim la liniște. Viteza acestor tranziții diferă semnificativ între tehnicile de ciupire și, uneori, între tehnicile stilistice.

Pentru a măsura cele două faze, acumulăm en-

ergia primelor zece armonici ale frecvenței fundamentale pentru fiecare cadru (i.e. unitate de timp) dintre onset și offset. Rezultă o curbă de înveliș care descrie variația amplitudinii în timp.

Panta de atac este calculată prin regresie liniară pe porțiunea dintre onset și momentul de maxim de pe curbă. O valoare mare indică un atac rapid și puternic, iar o valoare mică indică unul blând, specifice stilurilor slap și pick, respectiv finger-style.

Pentru panta de decădere, folosim tot regresie liniară, însă pe logaritmul amplitudinii, deoarece această fază urmează aproximativ o curbă exponențială. Obținem o pantă care indică viteza cu care sunetul se stinge. Spre exemplu, tehnica muted produce o decădere rapidă.

Aici calculăm următoarele caracteristici: timpul de atac în scară logaritmică, centroidul temporal al învelișului, panta de atac θ_1 din modelul liniar

$$a(n) \approx \theta_1 n + \theta_0, \tag{42}$$

și panta de decădere δ_1 din modelul exponențial

$$a(n) \approx a(n_{\text{Peak}})\, e^{-\delta_1 n}. \tag{43}$$

În plus, calculăm statistici pe învelișul de amplitudine separat pe atac și decădere (medie, varianță, minim, maxim, mediană, asimetrie și boltire), precum și aceleași statistici pe derivata temporală a învelișului.

2) Structura armonică: A doua categorie de indici descrie conținutul spectral al sunetului, mai precis raportul dintre frecvența fundamentală și frecvențele armonice ale sale. Fundamentala determină înălțimea în sine a notei, pe când armonicele determină timbrul acesteia.

Pentru armonicele de la a doua la a zecea, calculăm raportul între amplitudinea armonicii și amplitudinea fundamentalei, la momentul de amplitudine maximă a notei. Această valoare indică dacă armonicele au o prezență puternică în sunet.

De asemenea, măsurăm deviația fiecărei armonici față de frecvența sa teoretică. Aceste deviații naturale sunt relevante în diferențierea tehnicilor precum bend-ul sau slide-ul, unde tensiunea corzii se modifică în timpul notei.

Din distribuția acestor valori extragem statistici agregate: minim, maxim, medie și mediană, varianță, asimetrie și boltire. Astfel, condensăm informația într-un set compact de valori ușor utilizabile de clasificatoare.

Aici calculăm următoarele caracteristici: pentru fiecare armonică h la momentul de maxim, folosim raportul relativ

$$\chi_{a,\text{rel}}(h) = \frac{a_h(n_{\text{Peak}})}{a_0(n_{\text{Peak}})}, \quad h = 1, \ldots, 9, \tag{44}$$

precum și panta de decădere spectrală pe axa armonicilor

$$a_h \approx \gamma_1 h + \gamma_0. \tag{45}$$

Pentru deviațiile de frecvență folosim

$$\chi_{\Delta,f}(h) = \frac{\hat{f}_h(n_{\text{Peak}}) - f_h(n_{\text{Peak}})}{f_h(n_{\text{Peak}})}. \tag{46}$$

Pentru acești vectori se calculează statisticile: minim, maxim, medie, mediană, varianță, asimetrie și boltire.

3) Descriptori spectrali globali: Pe lângă analiza individuală a armonicelor, calculăm descriptori care caracterizează spectrul per ansamblu. Centroidul spectral indică centrul de greutate al spectrului și este corelat cu strălucirea sunetului. Răspândirea spectrală (distribuția frecvențelor sunetului) măsoară dispersia energiei. Rolloff-ul spectral indică frecvența sub care se află 85% din energie. Factorul de creastă compară valoarea maximă cu media, iar panta spectrală arată cum variază energia pe axa frecvențelor.

Calculăm acești descriptori atât pentru faza de atac, cât și pentru faza de decădere, iar pentru fiecare obținem media și varianța.

Aici calculăm următoarele caracteristici: centroid spectral, răspândire spectrală, rolloff la 85% din energie, factor de creastă, derivata temporală a factorului de creastă și panta spectrală. Acestea sunt calculate pe fiecare cadru din atac și decădere, iar pentru fiecare dintre ele se extrag statisticile: minim, maxim, medie, mediană, varianță, asimetrie și boltire, separat pentru cele două faze ale notei.

4) Tristimulus (Jensen, 1999): Tristimulusul este un concept din psihoacustică care împarte spectrul sonor în trei benzi de frecvențe armonice. Prima bandă conține energia fundamentalei și este corelată cu claritatea notei. A doua bandă grupează armonicele de la a doua până la a patra și indică căldura sunetului. Ultima bandă conține restul armonicelor și determină strălucirea sunetului.

Aici calculăm cele trei componente ale tristimulusului:

$$T_1 = \frac{a_1}{\sum_h a_h}, \quad T_2 = \frac{a_2 + a_3 + a_4}{\sum_h a_h}, \quad T_3 = \frac{\sum_{h\geq 5} a_h}{\sum_h a_h}. \tag{47}$$

precum și neregularitatea spectrală, definită ca suma diferențelor dintre armonice vecine, normalizată la energia totală. Pentru aceste mărimi se calculează aceleași statistici agregate pe atac și decădere.

5) Zgomotozitatea și subarmonici: Energia din spectru este însă provenită și din zgomot. Măsurăm zgomotozitatea ca fracția de energie care nu se află în vecinătatea armonicelor, relevantă în tehnicile slap și muted, datorită lovirii corzilor de taste, respectiv degete, producând valori ridicate ale acestor măsurători.

Pentru detectarea armonicelor naturale (flageolet), verificăm dacă există energie la subarmonicele frecvenței fundamentale (jumătate, treime, pătrime etc. din aceasta), care apar atunci când coarda este stinsă în locuri specifice.

Aici calculăm zgomotozitatea conform

$$\chi_{\text{Noisiness}} = \frac{1}{N} \sum_n \frac{\sum_k M_R(k,n)}{\sum_k M(k,n)}, \tag{48}$$

unde M_R este spectrul după eliminarea vârfurilor armonice cu ajutorul unui template spectral. Pentru subarmonici, calculăm caracteristicile

$$\chi_{\text{sub},m}, \quad m = 2, \ldots, 7, \tag{49}$$

ca rapoarte energetice între spectru și template-uri acordate la frecvențele f_0/m. În mod similar, calculăm patru valori de tip string likelihood, folosind template-uri acordate la frecvențele corzilor libere E, A, D și G.

6) Caracteristici de modulație a frecvenței: Această ultimă categorie conține indici care descriu tehnici expresive ce implică variația înălțimii notei în timp. Vibrato produce o oscilație periodică, bend-ul o creștere graduală, iar slide-ul o glisare (de obicei în trepte, datorită tastelor) între note.

Analizăm conturul frecvenței fundamentale și calculăm frecvența de modulație, amplitudinea modulației exprimată în cenți și progresia înălțimii prin compararea începutului și sfârșitului notei.

Rezultă, în urma acestor etape, un vector de aproximativ 212 de caracteristici pentru fiecare notă detectată.

Aici calculăm frecvența medie de modulație obținută din autocorelația conturului $f_0(n)$, numărul de sferturi de perioadă ale modulației, amplitudinea acesteia

$$\chi_{\text{lift}} = 1200 \log_2 \left(\frac{\max f_0(n)}{\min f_0(n)} \right), \tag{50}$$

și progresia de pitch

$$\chi_{\text{prog}} = 1200 \log_2 \left(\frac{\overline{f_0}^{\text{final}}}{\overline{f_0}^{\text{inițial}}} \right). \tag{51}$$

H. Clasificare

După analizarea notelor și obținerea vectorului de caracteristici, utilizăm trei SVM-uri independente pentru a decide stilul de interpretare, tehnica expresivă și coarda pe care a fost cântată nota. Stilul de interpretare îl împărțim în patru clase: Finger, Pick, Muted și Slap (articolul original și datasetul au două clase de slap, pe care noi le contopim: slap cu degetul mare și prin ciupire). Tehnica expresivă are șase clase: Normal, Vibrato, Bending, Slide, Harmonics, Dead-note (din nou, simplificăm prin combinarea unor clase precum slide up și down, bend mai mic sau mai mare etc.). Coarda poate fi una din patru standard: E, A, D, G. Fiecare clasificator este identic, în afară de clase: normalizare cu StandardScaler, selecție de caracteristici cu SelectKBest (primele 60) și clasificare cu kernel RBF. Hiperparametrii au fost optimizați prin GridSearch cu validare încrucișată: a rezultat alegerea ponderării claselor, C = 10, gamma calculată automat cu opțiunea scale. Folosim datasetul IDMT-SMT-BASS, din care reușim să parsăm totalul de 5306 înregistrări. Modelele sunt antrenate pe acesta și salvate pe disc pentru evaluare și inferență.

I. Calcularea poziției pe grif

Pentru a genera tabulatura, determinăm poziția pe grif folosind nota MIDI și coarda prezisă de clasificatorul menționat mai sus. Fret-ul este diferența dintre frecvența de bază a corzii și nota MIDI găsită. Pentru basul standard cu 4 corzi, acordat în EADG, notele MIDI ale corzilor sunt, respectiv, 28, 33, 38, 43. Dacă tehnica detectată este Dead-note, returnăm -1, deoarece nu se produce un pitch propriu-zis, ci mai degrabă o percuție. Rezultatul final este o listă de evenimente descrise prin: momentul onset-ului, pitch-ul MIDI, stilul, tehnica expresivă, coarda și fret-ul.

J. Rezumat pipeline:

Sistemul extrage aproximativ 212 de indici caracteristici per notă, grupați în categoriile menționate anterior. Clasificarea este realizată cu trei clasificatoare SVM independente, iar poziția pe grif este estimată din măsurătorile de pitch și estimarea corzii.

V. Evaluare

Evaluarea este realizată în două moduri distincte: pe datasetul de note izolate și pe cel de linii de bas complete. Pe datasetul IDMT-SMT-BASS, folosim validare încrucișată stratificată, cu 5 fold-uri, și am obținut următoarele rezultate ale acurateții:

Metrică	**Acuratețe**
Stilul de interpretare / tehnica de ciupire	98%
Tehnica expresivă	94%
Coarda	87%

Pe datasetul cu linii complete, pentru 17 linii, valorile obținute sunt:

Metrică	**Valoare**
F-measure detecție note	0.793
Pitch	95% acc.
Stilul de interpretare / tehnica de ciupire	46% acc.
Tehnica expresivă	48% acc.
Coarda	29% acc.

VI. Concluzii si directii viitoare

În această lucrare am reprodus arhitectura propusă de Abeßer și Schuller pentru transcrierea chitarei bas și am evaluat fidelitatea implementării noastre folosind același set de date ca în lucrarea originală. Rezultatele obținute ne permit să formulăm câteva concluzii clare.

A. Concluzii

- **Limitarea setului de date afectează direct acuratețea detecției notelor.** Deoarece dataset-ul disponibil nu conține variații de durată, nu am putut reproduce fidel evaluarea originală. În consecință, performanța noastră la detecția notelor este mai scăzută: **recall = 0.878, precision = 0.723, F-measure = 0.793**, comparativ cu **0.897, 0.908, 0.901** raportate în lucrarea de referință. Această diferență este explicabilă prin lipsa variabilității temporale, care reduce robustețea detecției offset-urilor.
- **Detecția onset-urilor este puternic influențată de pragul ales.** Pragul recomandat în lucrarea originală (0.20) a generat în datele noastre un număr ridicat de detecții false: **1491 onset-uri detectate, dintre care 569 false**. Ajustarea pragului la 0.30 a redus semnificativ supra-detectarea, obținând **1141 onset-uri detectate, dintre care 279 false**. Această îmbunătățire a stabilității vine însă cu un compromis minor în acuratețea estimării înălțimii: **pitch accuracy scade de la 0.954 (prag 0.20) la 0.943 (prag 0.30)**. Rezultatele sugerează că un prag mai ridicat oferă un echilibru mai bun între robustețea detecției și calitatea estimării lui f_0.

- **Pipeline-ul nostru îmbunătățește semnificativ clasificarea tehnicilor de interpretare.** Datorită ajustării hiperparametrilor și introducerii unor caracteristici timbrale suplimentare, am obținut rezultate superioare celor raportate în lucrarea originală pentru:
  - **stilul de ciupire**: 0.46 vs. 0.282,
  - **stilul de expresie**: 0.48 vs. 0.2417.

  Acest lucru sugerează că modelul nostru captează mai bine variațiile timbrale asociate tehnicilor de interpretare.
- **Estimarea corzii rămâne o sarcină dificilă.** Pentru clasificarea corzii, performanța noastră este mai scăzută decât cea raportată în lucrarea originală: **0.29 vs. 0.495**. Analiza noastră indică faptul că această degradare este cauzată de sensibilitatea ridicată a clasificatorului la inarmonicitate și la structura armonicilor, care sunt mai greu de estimat în notele scurte și în registrele joase.
- **Cross-validation confirmă că modelul nostru generalizează bine.** Atunci când aplicăm validare încrucișată, obținem rezultate foarte apropiate de cele din lucrarea originală:
  - tehnică de interpretare: **0.98 vs. 95.6%**,
  - expresivitate: **0.94 vs. 94.7%**,
  - coardă: **0.87 vs. 87.8%**.

  Acest lucru arată că, în condiții controlate, pipeline-ul nostru reproduce fidel comportamentul modelului original.

B. Direcții viitoare

- **Înlocuirea SVM cu modele moderne de învățare profundă.** Rețelele convoluționale sau arhitecturile bazate pe spectrograme logaritmice ar putea îmbunătăți semnificativ clasificarea tehnicilor de interpretare și a poziției pe grif.
- **Îmbunătățirea detecției onset-urilor prin metode adaptive.** Un prag fix funcționează bine doar pentru anumite stiluri de interpretare. Un sistem adaptiv, bazat pe energie sau pe variația locală a armonicilor, ar putea reduce erorile în stilurile slap și muted.
- **Modelarea temporală a tehnicilor expresive.** Tehnici precum vibrato, bending sau slide au semnături temporale clare. Integrarea unor modele secvențiale (HMM, LSTM) ar putea crește acuratețea clasificării.
- **Extinderea setului de caracteristici timbrale.** Caracteristici precum fluxul spectral, entropia spectrală sau descriptorii de zgomot ar putea ajuta la diferențierea mai bună a tehnicilor slap și dead-note.
- **Evaluarea pe date reale, necurățate.** Setul de date folosit este controlat și curat. Testarea pe înregistrări reale, cu zgomot și variații de microfon, ar oferi o imagine mai realistă asupra robusteții metodei.

În concluzie, implementarea noastră reproduce în mare parte comportamentul pipeline-ului original, dar evidențiază și limitele acestuia. Rezultatele sugerează că, deși arhitectura propusă de Abeßer

și Schuller este solidă, există spațiu semnificativ pentru îmbunătățiri, în special în ceea ce privește clasificarea tehnicilor expresive și estimarea poziției pe grif.

REFERINȚE

[1] J. Abeßer and G. Schuller, "Instrument-centered music transcription of solo bass guitar recordings," in *IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, VOL. 25, NO. 9, SEPTEMBER 2017*. IEEE, 2017, pp. 1–9.

[2] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription: challenges and future directions," *Journal of Intelligent Information Systems*, vol. 41, no. 3, pp. 407–434, 2013.

[3] S. W. Hainsworth and M. D. Macleod, "Automatic bass line transcription from polyphonic music," in *Proceedings of the International Computer Music Conference*. La Habana, Cuba: ICMC, 2001, pp. 431–434.

[4] M. P. Ryynänen and A. Klapuri, "Automatic transcription of melody, bass line, and chords in polyphonic music," *Computer Music Journal*, vol. 32, pp. 72–86, 2008.

[5] Y. Uchida and S. Wada, "Melody and bass line estimation method using audio feature database," in *Proceedings of the IEEE International Conference on Signal Processing, Communications and Computing*. IEEE, 2011, pp. 1–6.

[6] E. Tsunoo, N. Ono, and S. Sagayama, "Musical bass-line pattern clustering and its application to audio genre classification," in *Proceedings of the 10th International Society for Music Information Retrieval Conference*. Kobe, Japan: ISMIR, 2009, pp. 219–224.

[7] C. Dittmar, K. Dressler, and K. Rosenbauer, "A toolbox for automatic transcription of polyphonic music," in *Proceedings of the Audio Mostly Conference*. Audio Mostly, 2007, pp. 58–65.

[8] M. Goto, "A real-time music-scene-description system—predominant-f0 estimation for detecting melody and bass lines in real-world audio signals," *Speech Communication*, vol. 43, no. 4, pp. 311–329, 2004.

[9] E. Tsunoo, G. Tzanetakis, N. Ono, and S. Sagayama, "Beyond timbral statistics: Improving music classification using percussive patterns and bass lines," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 1003–1014, 2011.

[10] J. Salamon and E. Gómez, "A chroma-based salience function for melody and bass line estimation from music audio signals," in *Proceedings of the 6th Sound and Music Computing Conference*. Porto, Portugal: SMC, 2009, pp. 23–25.

[11] M. Mauch and S. Dixon, "pyin: A fundamental frequency estimator using probabilistic threshold distributions," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. Florence, Italy: IEEE, 2014, pp. 659–663.

[12] J. Abeßer and G. Schuller, "Instrument-centered music transcription of bass guitar tracks," in *Proceedings of the AES 53rd Conference on Semantic Audio*. London, U.K.: AES, 2014, pp. 166–175.

[13] J. Abeßer, "Automatic transcription of bass guitar tracks applied for music genre classification and sound synthesis," Ph.D. dissertation, Ilmenau University of Technology, Ilmenau, Germany, 2014.

[14] A. Loscos, Y. Wang, and W. J. J. Boo, "Low level descriptors for automatic violin transcription," in *Proceedings of the 7th International Conference on Music Information Retrieval*. Victoria, Canada: ISMIR, 2006, pp. 164–167.

[15] L. S. Pardue, C. Harte, and A. P. McPherson, "A low-cost realtime tracking system for violin," *Journal of New Music Research*, vol. 44, no. 4, pp. 305–323, 2015.

[16] J. Yin, Y. Wang, and D. Hsu, "Digital violin tutor: An integrated system for beginning violin learners," in *Proceedings of ACM Multimedia*. Singapore: ACM, 2005, pp. 976–985.

[17] Y.-L. Chen, T.-M. Wang, W.-H. Liao, and A. W. Y. Su, "Analysis and trans-synthesis of acoustic bowed-string instrument recordings—a case study using bach cello suites," in *Proceedings of the 14th International Conference on Digital Audio Effects*. Paris, France: DAFx, 2011, pp. 63–67.

[18] X. Fiss and A. Kwasinski, "Automatic real-time electric guitar audio transcription," in *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing*. Praha, Czech Republic: IEEE, 2011, pp. 373–376.

[19] C. Kehling, J. Abeßer, C. Dittmar, and G. Schuller, "Automatic

tablature transcription of electric guitar recordings by estimation of score and instrument-related parameters," in *Proceedings of the 17th International Conference on Digital Audio Effects*. Erlangen, Germany: DAFx, 2014, pp. 1–8.

[20] V. Emiya, R. Badeau, and B. David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2010.

[21] E. Maestre, M. Blaauw, J. Bonada, E. Guaus, and A. Pérez, "Statistical modeling of bowing control applied to violin sound synthesis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 4, pp. 855–871, 2010.

[22] A. P. Carrillo, J. Bonada, E. Maestre, E. Guaus, and M. Blaauw, "Performance control driven violin timbre model based on neural network," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 1007–1021, 2012.

[23] C. Erkut, M. Karjalainen, and M. Laurson, "Extraction of physical and expressive parameters for model-based sound synthesis of the classical guitar," in *Proceedings of the 108th Audio Engineering Society Convention*. AES, 2000, pp. 19–22.

[24] E. Guaus, T. Ozaslan, E. Palacios, and J. L. Arcos, "A left hand gesture caption system for guitar based on capacitive sensors," in *Proceedings of the 10th International Conference on New Interfaces for Musical Expression*. Sydney, Australia: NIME, 2010, pp. 238–243.

[25] L. Reboursière, O. Lähdeoja, T. Drugman, S. Dupont, C. Picard-Limpens, and N. Riche, "Left and right-hand guitar playing techniques detection," in *Proceedings of the International Conference on New Interfaces for Musical Expression*. Ann Arbor, MI, USA: NIME, 2012, pp. 1–4.

[26] T. H. Ozaslan, E. Guaus, E. Palacios, and J. L. Arcos, "Attack based articulation analysis of nylon string guitar," in *Proceedings of the 7th International Symposium on Computer Music Modeling and Retrieval*. Málaga, Spain: CMMR, 2010, pp. 285–298.

[27] C. Traube and J. O. Smith, "Extracting the fingering and the plucking points on a guitar string from a recording," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. New Paltz, NY, USA: IEEE, 2001, pp. 2–5.

[28] I. Barbancho, C. de la Bandera, A. M. Barbancho, and L. J. Tardón, "Transcription and expressiveness detection system for violin music," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. Taipei, Taiwan: IEEE, 2009, pp. 189–192.

[29] Y.-P. C. Chen, L. Su, and Y.-H. Yang, "Electric guitar playing technique detection in real-world recordings based on f0 sequence pattern recognition," in *Proceedings of the 16th International Society for Music Information Retrieval Conference*. ISMIR, 2015, pp. 708–714.

[30] M. Lagrange and S. Marchand, "Estimating the instantaneous frequency of sinusoidal components using phase-based methods," in *Journal of the Audio Engineering Society*, vol. 55, no. 5. AES, 2007, pp. 385–399.