APPRENTISSAGE PROFOND POUR LA RECONNAISSANCE EN TEMPS REEL DES MODES DE JEU INSTRUMENTAUX

Jean-Francois Ducher, Philippe Esling

APPRENTISSAGE PROFOND POUR LA RECONNAISSANCE EN TEMPS REEL DES MODES DE JEU INSTRUMENTAUX

Jean-François DUCHER
CICM – MUSIDANSE – Université Paris 8/
IRCAM (UMR9912 STMS)
ducher@ircam.fr

Philippe ESLING
IRCAM (UMR9912 STMS)
esling@ircam.fr

RÉSUMÉ

Au cours des dernières années, l'apprentissage profond s'est établi comme la nouvelle méthode de référence pour les problèmes de classification audio et notamment la reconnaissance d'instruments. Cependant, ces modèles ne traitent généralement pas la classification de modes de jeux avancés, question pourtant centrale dans la composition contemporaine. Les quelques études réalisées se cantonnent à une évaluation sur une seule banque de sons, dont rien n'assure la généralisation sur des données réelles.

Dans cet article, nous étendons les méthodes de l'état de l'art à la classification de modes de jeu instrumentaux en temps réel à partir d'enregistrements de solistes. Nous montrons qu'une combinaison de réseaux convolutionnels (CNN) et récurrents (RNN) permet d'obtenir d'excellents résultats sur un corpus homogène provenant de 5 banques de sons. Toutefois, leur performance s'affaiblit sensiblement sur un corpus hétérogène, ce qui pourrait indiquer une faible capacité à généraliser à des données réelles. Nous proposons des pistes pour résoudre ce problème. Enfin, nous détaillons plusieurs utilisations possibles de nos modèles dans le cadre de systèmes interactifs.

1. INTRODUCTION

Depuis le début du XXème siècle, la volonté d'innover dans le domaine du timbre a conduit les compositeurs à participer à la conception de nouveaux instruments et à utiliser l'électronique dans leurs œuvres. Cependant, l'*instrumentarium* issu du grand orchestre symphonique romantique a été majoritairement conservé et cette recherche s'est portée principalement vers le geste instrumental qui permettrait de réaliser le son souhaité. Qu'ils soient interprètes ou improvisateurs, les instrumentistes se sont ensuite réapproprié ces modes de jeu qui se sont diffusés hors du champ de la musique contemporaine.

Parallèlement, le champ des systèmes musicaux interactifs s'est considérablement développé. Qu'il s'agisse de systèmes de suivi de partition en temps réel, de systèmes d'improvisation, ou de dispositifs de musique mixte, la plupart s'appuient sur des descripteurs plus ou moins complexes pour analyser le flux audio qui les alimente [21]. Ces systèmes pourraient donc bénéficier de modules permettant de reconnaître en temps réel le mode de jeu utilisé par l'instrumentiste.

Appliqué à des musiques de tradition orale ou plus largement des enregistrements pour lesquels aucune partition n'est disponible, un module de classification des modes de jeu constituerait également un outil utile pour la transcription et l'analyse musicologique [6].

Un effort de recherche considérable a été déployé par la communauté du Music Information Retrieval (MIR) pour résoudre un ensemble de problèmes connexes à l'aide d'algorithmes d'apprentissage profond [4]. Ces algorithmes peuvent permettre notamment de distinguer le timbre des différents instruments à partir du signal audio provenant d'échantillons de notes isolés, d'enregistrements de solistes ou de textures polyphoniques [14, 20]. En revanche, la reconnaissance des modes de jeu instrumentaux (MJI) a fait l'objet de peu d'études, notamment du fait du faible nombre de banques de sons suffisamment exhaustives, en particulier s'agissant des techniques contemporaines. A fortiori, il n'existe pas à notre connaissance de bases d'enregistrements de solistes labellisés qui prenne en compte ces techniques.

Signe de l'intérêt de la communauté MIR pour le sujet, Lostanlen et al. [14] ont récemment proposé un dispositif de reconnaissance de MJI susceptible de fonctionner sur un grand nombre d'instruments. Néanmoins, leur étude porte uniquement sur des échantillons de notes isolées provenant d'une seule banque de sons.

Dans cet article, nous nous attaquons pour la première fois à la réalisation d'un classificateur de MJI en temps réel à partir d'enregistrements de solistes. La contrainte du temps réel implique que notre système soit réactif à un changement de mode de jeu de l'instrumentiste. Nous excluons notamment une segmentation du flux audio en amont du système [19] qui nous permettrait de nous ramener au problème de la classification à partir de notes isolées. Nos expériences se concentrent sur le violoncelle mais les enjeux techniques et méthodologiques restent identiques pour d'autres instruments, sous réserve qu'ils soient représentés dans un nombre suffisant d'échantillons.

Pour entraîner notre classificateur, nous avons produit un large corpus de données synthétiques

labellisées à partir de 5 banques de sons de MJI et de leurs séquenceurs propriétaires. Nous réalisons deux expériences qui s'appuient sur des réseaux de neurones à convolution (CNN) appliqués à des mel-log-spectrogrammes du flux audio, représentant l'état de l'art en matière de reconnaissance instrumentale pour des textures polyphoniques [9,20].

Dans la première expérience, nous combinons le CNN profond avec une couche de neurones dite intégralement connectée et effectuons l'apprentissage supervisé sur notre corpus de données synthétiques. Nous présentons les résultats de cette expérience lorsque le jeu de données de test et d'apprentissage sont choisis au hasard dans ce corpus, traité de façon homogène. Nous comparons ces résultats avec ceux d'une variante où un réseau de neurones récurrents (RNN) remplace la couche intégralement connectée : cette couche de RNN apporte théoriquement une faculté d'oubli au réseau et donc une plus grande réactivité aux changements de mode en temps réel.

Notre enjeu étant de généraliser à des enregistrements réels de solistes, nous adaptons au besoin d'évaluation de notre système la méthodologie « minus-1db ». Livshin [13] a proposé cette méthodologie pour mesurer la capacité de généralisation de classificateurs de timbres instrumentaux entraînés à partir de bases de données d'échantillons. Nous en présentons les résultats pour nos deux expériences et discutons les limites de ce paradigme d'évaluation. Nous évaluons également les résultats de nos modèles face aux contraintes temps réel.

Enfin, nous détaillons des cas d'applications possibles de notre système dans un cadre créatif de composition musicale s'appuyant sur des systèmes intéractifs.

2. ETAT DE L'ART

2.1. Apprentissage profond et MIR

Suite aux succès rencontrés dans leurs nombreuses applications dans le domaine de l'image [8], les techniques d'apprentissage profond ont été rapidement adoptées par la communauté MIR [4]. Au lieu d'utiliser un ensemble de descripteurs audio complexes pour alimenter des algorithmes d'apprentissage machine, ces techniques s'appuient sur des représentations basiques du signal audio, et laissent l'algorithme apprendre par lui-même les représentations les plus adaptées à une tâche particulière. Parmi ces méthodes, les réseaux de neurones à convolution (CNN) et les réseaux récurrents (RNN) font partie des architectures les plus populaires. Nous redirigeons les lecteurs intéressés vers [4] pour un inventaire détaillé des architectures proposées et leurs applications musicales.

2.1.1. Réseaux de neurones à convolution

Les CNN sont construits par superposition de couches de convolution qui appliquent un ensemble de filtres locaux à travers les dimensions de la donnée d'entrée. Ainsi, ces réseaux permettent de détecter des motifs en calculant des corrélations locales grâce à un noyau (*kernel*) dont les paramètres sont déterminés par le processus d'apprentissage. La sortie d'une telle couche, appelée carte de convolution (*feature map*) est souvent passée à une couche de *pooling* qui réduit les dimensions de cette carte en la sous-échantillonnant (en général en prenant le maximum[1] ou la moyenne des valeurs sur un ensemble de pixels), donnant ainsi au résultat une propriété d'invariance à la translation. L'empilement de couches de convolution et de *pooling* à diverses échelles permet de détecter des motifs plus grands et plus complexes. Ces mécanismes sont résumés dans la figure 1.

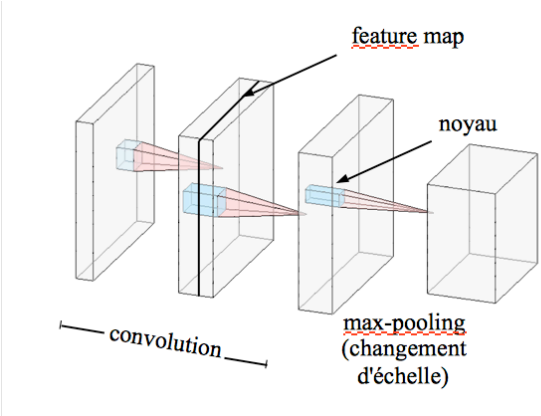

Figure 1. Schéma général d'un CNN

Pour réduire la charge de calcul, les premiers travaux se sont appuyés sur les *Mel-Frequency Cepstral Coefficients* (MFCC) qui effectuent une compression des données en prenant la transformée en cosinus discrète du logarithme de l'amplitude des mel-spectrogrammes [1,16]. Cette opération s'effectue au détriment du caractère local des motifs apparaissant dans les spectrogrammes. De fait, cette approche a été progressivement remplacée par des représentations plus simples [4], comme la transformée de Fourier à court terme, les mel-spectrogrammes voire même le signal audio brut.

Les applications actuelles des CNN dans le domaine du MIR vont de la reconnaissance d'accords, la détection d'attaque, à la classification d'instruments ou de genre [4,20].

2.1.2. Réseaux de neurones récurrents

Les RNN ont été développés dans le but de prédire ou classifier des séquences temporelles. Leurs couches

[1]On parle alors de *max-pooling*.

cachées font intervenir des connections récurrentes d'un pas temporel au suivant, transportant ainsi de l'information à travers les différents états temporels (voir la figure 2). Des portes (*gates*) ont été introduites pour contrôler ce flux d'information.

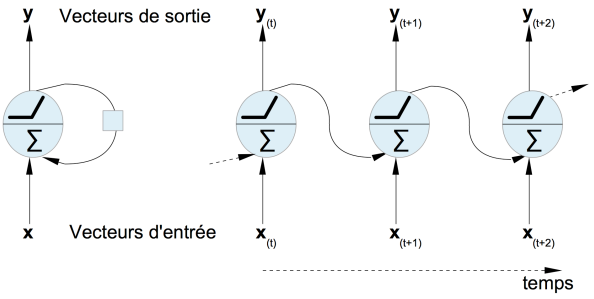

Figure 2. Représentation d'un réseau récurrent élémentaire combinant linéairement son entrée avec l'état au pas temporel précédent puis le soumettant à une fonction d'activation non linéaire. A droite le dépliement temporel du mécanisme.

Les réseaux à portes, avec des cellules de type *Long Short-Term Memory* (LSTM) ou leur version simplifiée les *Gated Recurrent Units* (GRU), ont connu un vif succès dans des tâches de reconnaissance du langage ou de traduction [8]. Dans le domaine musical, ces modèles ont principalement été utilisés pour des problèmes de prédiction à court terme comme la détection de chant ou la transcription instrumentale. Enfin, certaines combinaisons de RNN et de CNN ont été introduites pour des problématiques de classification musicale [4].

2.2. Reconnaissance des timbres instrumentaux

Les premiers travaux de recherche sur le timbre instrumental procédaient à partir d'échantillons de notes isolées jouées avec des techniques « ordinaires ». Dans la plupart des études, les expériences étaient menées avec une seule banque de sons, avec peu de variabilité intra-classe due au modèle de l'instrument, au jeu propre à l'instrumentiste ou bien à l'environnement de prise de son. Même si ces méthodes obtiennent d'excellents résultats sur ces bases, rien ne garantit leur succès dans des conditions d'utilisation réelles, voire même simplement sur une autre base de données. En effet, une revue détaillée des problématiques de généralisation par Livshin [13] a montré qu'il n'y avait aucun moyen d'évaluer la précision d'un classificateur sur des sons nouveaux à partir de ses performances mesurées sur des données provenant d'une seule banque de sons.

Livshin a proposé d'utiliser plusieurs banques de sons indépendantes et de tester sur chacune le classificateur entraîné sur l'ensemble des autres banques réunies. Cette méthodologie qu'il a nommée *minus-1db* fournit des indications plus fiables quant à la capacité du classificateur à généraliser. En partitionnant le jeu de données de façon hétérogène entre apprentissage et test, elle est notamment plus exigeante que la *k-fold validation*, autre bonne pratique de type *cross-validation* souvent mise en œuvre en apprentissage machine, qui repose sur une partition aléatoire (donc homogène) des données. Quant à la classification instrumentale à partir d'enregistrements de solistes, une transposition possible de la méthodologie *minus-1db* consiste à prendre comme base de tests les enregistrements venant d'un CD et d'entraîner le réseau sur l'ensemble des autres (*leave-1CD-out*).

Livshin a également démontré dans le cas de son classificateur[2] qu'enrichir la base d'apprentissage avec des échantillons provenant d'autres banques de sons contribuait à une meilleure généralisation.

Seules deux études suivent les principes méthodologiques de Livshin et peuvent donc prétendre à l'état de l'art actuel sur cette problématique. Premièrement, un classificateur construit sur une Machine à Vecteur de Support (SVM) appliqué à des Champs Récepteurs Spectro-Temporels a été proposé par [19]. Entraîné sur la base de données RWC, ce modèle a atteint 98.7% de précision pour classer 6 instruments à partir d'échantillons de notes isolées. La robustesse du modèle a été évaluée sur une base propriétaire de soli instrumentaux segmentés avec une méthode liée à l'harmonicité du signal. Sur cette même base, la performance du modèle était meilleure que lorsque la SVM était appliquée à un jeu de descripteurs spectraux MPEG-7. Enfin, entraîné sur la base de soli avec la méthodologie *leave-1CD-out*, les taux de précision atteignaient encore 88.1%. Deuxièmement, Lostanlen et Cella [15] ont utilisé deux bases de données indépendantes de soli instrumentaux pour entraîner et tester un CNN profond s'appuyant sur le CQT du signal audio pour discriminer 8 instruments. En optimisant leur stratégie de convolution, des précisions moyennes de l'ordre de 74% sont obtenues, contre 61.4% pour une forêt d'arbres décisionnels appliqué à un large ensemble de descripteurs audio.

2.2.1. Textures polyphoniques

S'agissant de reconnaître les instruments prédominants dans un ensemble polyphonique, un CNN profond appliqué à des log-mel-spectrogrammes, construit par Han et al. [9], a atteint l'état de l'art . Cette étude a été réalisée avec deux sous-ensembles indépendants de la base de données IRMAS, respectivement pour l'apprentissage et le test.

Les efforts de la communauté MIR pour construire des bases de données de textures polyphoniques qui soient ouvertes, suffisamment importantes et

[2] Une *analyse discriminante linéaire* qui opérait une sélection et une pondération sur un ensemble considérable de descripteurs audio possibles, combinée avec un classificateur de type *k-proches voisins*.

correctement labellisées se poursuivent également avec l'initiative Open-MIC [10].

2.3. Classification des Modes de Jeu Instrumentaux

Des expériences de classification de MJI ont été réalisés sur la clarinette [17], la caisse claire [22] et la guitare électrique [3]. Les deux premières études procèdent à partir d'échantillons de notes séparées venant d'une seule base de données propriétaire, de sorte qu'aucune précision n'est apportée quant à la capacité du classificateur à généraliser. Chen et al. [3] se concentrent sur la détection dans les solos de guitare électrique de 5 techniques employées dans le rock qui ont toutes en commun le fait d'affecter le contour mélodique (*bend*, *slide*, *vibrato*, *pull-off* et *hammer-on*). Cette caractéristique est au cœur de la conception de leur système[3], ce qui rend le rend difficile à généraliser à d'autres MJI.

Enfin, Lostanlen et al. [14] posent la question de la classification des MJI d'une façon transversale aux 16 instruments présents dans la banque de sons Studio-On-Line (SOL) avec laquelle ils travaillent. SOL contient des échantillons de notes isolées de 143 MJI différents. Le système d'interrogation par l'exemple (*query-by-example*) mis au point par les auteurs s'appuie sur une variante de l'algorithme des k-proches voisins où la métrique utilisée fait l'objet d'un apprentissage. Appliqué à un ensemble de descripteurs cepstraux, il parvient à un taux de précision au rang 5 de 61% mais rien ne permet d'évaluer sa capacité de généralisation.

3. DEFINITION DE LA TACHE PROPOSEE

3.1. Définition théorique des modes de jeu

Au fur et à mesure que les compositeurs testaient méthodiquement l'effet de la gestuelle de l'instrumentiste sur tous les paramètres physiques contribuant à l'excitation, la vibration et la résonance de l'instrument, et en définitive, sur le son produit, les MJI sont devenus des phénomènes multi-dimensionnels et complexes.

Prenant l'exemple du violoncelle, il conviendrait de distinguer en théorie :

i) la nature de l'excitateur, qui peut être l'archet, avec le crin, le bois ou un mix des deux avec une proportion variable, le doigt ou l'ongle (*pizz*.), la main (frappe), ou bien encore tout autre autre excitateur *exotique* ;

ii) la nature et la durée de l'interaction avec l'excitateur : on doit distinguer entre des modes non entretenus (*pizzicato*, *con legno battuto*), ou au contraire entretenus plus ou moins longuement (*staccato*, *marcato*) ; l'utilisation du rebond de l'archet sur la corde peut induire une quasi-périodicité de l'interaction (*jeté)* ; l'instrumentiste peut aussi jouer sur la pression et la vitesse d'archet, avec à l'extrême, la technique du *flautando* ou au contraire l'écrasement des cordes ;

iii) la position de l'interaction : *sul tasto*, *sul ponticello*, voire sur le chevalet ou derrière ; cette position peut elle-même faire l'objet d'une dynamique temporelle, on pense par exemple au balayage dans les positions intermédiaires entre la touche et le chevalet qu'opère le *circular bowing ;*

iv) au-delà, la nature-même du vibrateur : notamment la corde choisie (I-IV) pour la réalisation du son ; corde normalement fixée en deux points, mais éventuellement effleurée en un troisième (harmoniques). La mobilité de la corde au chevalet peut être limitée par l'usage d'une sourdine ; elle peut faire l'objet d'une préparation qui en altère les caractéristiques ; le cordier, les chevilles, la caisse peuvent-être sollicités comme vibrateur accessoire (pizz. Bartok) ou principal (coups portés sur le corps de l'instrument).

v) enfin, le jeu sur les hauteurs qui s'opère en main gauche : *vibrato* vs. *non vibrato*, *glissandi*, trilles.

3.2. Modes retenus dans notre étude

Prendre en compte toute la complexité des MJI supposerait de labelliser un corpus entier de solos instrumentaux suffisant pour l'apprentissage et le test du classificateur. Cette opération très coûteuse à réaliser sort du cadre de cette étude.

Ainsi notre classificateur doit s'appuyer sur des banques de sons instrumentales existantes, nous contraignant à homogénéiser les choix et les définitions de MJI de ces différentes bases. En effet, on ne retrouve dans les différentes banques de sons disponibles qu'une fraction des possibilités techniques évoquées précédemment. Cette fraction dépend des besoins anticipés des utilisateurs que ciblent leurs éditeurs. Les MJI y sont présentés de façon discrétisée[4] et le plus souvent uni-dimensionnelle. Par exemple, on trouvera côte-à-côte des échantillons de trilles, d'harmoniques artificielles et de jeu *sul ponticello* sans que soient envisagées les diverses combinaisons de ces techniques. Enfin, elles utilisent une terminologie qui ne fait l'objet d'aucune standardisation. Un exemple simple de ces écarts de définition porte sur le niveau de *vibrato* qu'on trouverait dans une classe libellée *ordinario*, qui n'est pas la même d'une banque à l'autre.

[3] Un système de règles appliqué au contour de la note fondamentale permet d'identifier des segments susceptibles de relever de ces différentes techniques. Ces segments sont ensuite soumis à une SVM entraînée à reconnaître de façon binaire chaque technique à partir d'un large ensemble de descripteurs.

[4] À l'exception de Ircam Solo Instruments qui dispose de quelques échantillons de transitions entre modes.

En dépit de ces limitations évidentes, nous reproduisons cette classification dans notre recherche. Nous avons identifié 5 banques de sons indépendantes en termes d'instrumentistes et de contexte de prise de son : EastWest Quantum Leap (EWQL), Vienna String Library (VSL), IRCAM Solo Instruments (ISI), Virtual Orchestra (VO) et ConTimbre (CONT). En croisant les MJI représentés dans au moins deux de ces banques de son, nous avons défini 18 classes. La table 1 présente les modes de jeu retenus et le nombre de banques de sons dans lesquelles elles apparaissent.

staccato, spiccato	2	con legno battuto	5
détaché	3	harmoniques	4
sustained vibrato	3	sul ponticello	4
marcato, sfz	3	sul pont. tremolo	3
sustained non vibrato	4	sul tasto	5
tremolo	4	sul tasto tremolo	2
trille	2	écrasé	3
pizzicato	5	con legno tratto	3
pizz. Bartok	4	hit on body	2

Table 1. Nombre de banques de sons dans lesquelles les 18 classes sont présentes au sein de notre corpus.

La tâche du classificateur consistera donc à attribuer à chaque séquence audio communiquée au système un label compris entre 0 et 17.

4. EXPERIENCES

4.1. Construction des bases de données

4.1.1. Principes de génération des séquences

Ayant en tête l'objectif de généralisation à de véritables enregistrements de solistes, nous avons généré à l'aide des séquenceurs propriétaires[5] des banques de sons, pilotés par des patchs MAX/MSP[6], un ensemble de séquences permettant de simuler un solo instrumental. Ceux-ci incluent ainsi de façon aléatoire des simples, doubles et tripes cordes, dans les limites de faisabilité de l'instrument et des tessitures autorisées par les différentes banques. Le simulateur réalise également des variations d'intensité sur les modes entretenus et des glissandi, lorsqu'ils sont possibles dans le mode concerné.

On réservera dans la suite le terme de bases de données à ces ensembles de séquences, par opposition aux banques de sons[7] qui ont permis de les constituer.

[5]Par exemple : UVI Workstation pour ISI ou Vienna Instruments pour VSL.
[6]Disponibles sur demande.
[7]On conservera pour la base la dénomination de la banque de son d'origine : e.g. ISI, CONT.

4.1.2. Augmentation des données

Dans la perspective d'augmenter la robustesse du classificateur à des variations dans l'environment d'enregistrement et l'accord de l'instrument, nous avons généré des séquences avec un « la » de référence aléatoirement modifié dans un bande de 20Hz autour de 440Hz (par transposition des échantillons).

Nous avons également fait varier les niveaux de réverbération dans les séquenceurs et utilisé des échantillons enregistrés à différentes distances de l'instrument, lorsque la banque de sons le permettait.

L'augmentation se traduit par un quadruplement de la quantité de données qui atteint l'équivalent de 18 heures d'audio brut. Le signal audio retenu est la moyenne des canaux stéréo fournis par le séquenceur.[8]

4.2. Architecture et configuration du système

4.2.1. Pré-traitement des données

Suivant en cela la méthodologie de [9], nous avons réduit la fréquence d'échantillonnage du signal audio à 24kHz, estimant que l'information fréquentielle au dessus de 12kHz n'est pas nécessaire à la reconnaissance des modes de jeu du violoncelle.

Nous calculons le logarithme des amplitudes des Mel-spectrogrammes[9] sur 64 bandes et éliminons les fenêtres (*frames*) ayant une énergie faible et dont le rapport signal/bruit est dégradé. Enfin, nous normalisons l'entrée du réseau pour chaque instance des paramètres d'augmentation et chaque banque de sons.

Avant d'être passées au réseau, les données sont ensuite découpées en séquences correspondant à un nombre fixe de fenêtres. Plusieurs expériences ont été réalisées avec des tailles de séquences différentes pour trouver le meilleur compromis entre la charge de calcul (à l'entraînement comme en inférence) et la perte d'information liée à la sortie de l'attaque de la séquence analysée par le réseau. Dans la suite de cette étude, la taille des séquences est fixée à 60 *frames*, correspondant approximativement à 1.2 secondes d'audio.

4.2.2. Architecture du réseau

Au vu des ressemblances entre notre tâche et celle de reconnaissance d'instruments dominants dans des textures polyphoniques, nous suivons les principales caractéristiques de l'architecture de CNN présentée par [9], en adaptant la capacité à notre volume de données.

[8]L'exploitation séparée des deux canaux constitue une source potentielle d'augmentation des données.
[9]FFT ave une fenêtre de 2048 échantillons et des sauts de 512 échantillons.

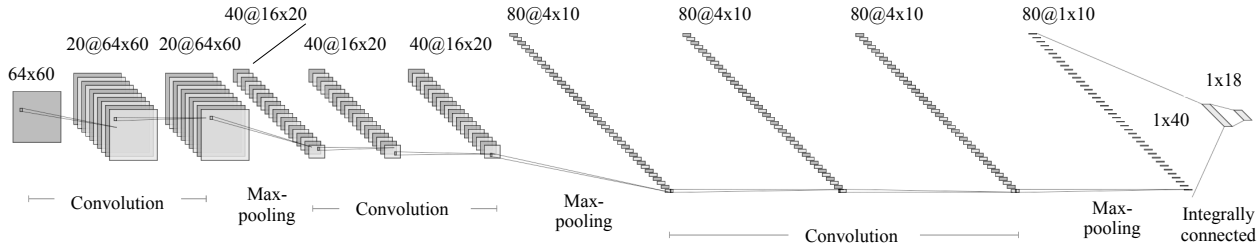

Figure 3. Architecture de réseau retenue pour l'expérience (1ère variante). Les dimensions des couches de convolution sont stipulées en *nombre de cartes @ pas fréquentiels* x *pas temporels.* Les filtres utilisés pour chaque module bi-couche sont respectivement 4x3, 4x2, 4x1.

Le CNN proposé est composé de 3 modules de convolution à des échelles croissantes qui transforment les séquences de 64 valeurs du mel-spectrogramme pour 60 *frames* en 80 descripteurs calculés tous les 6 *frames* (soit environ 126ms). Chaque module est constitué de deux couches de convolution (avec normalisation par batch et fonctions d'activation ReLU[10]), suivies d'une couche de *max-pooling*. La technique de *dropout* consistant à « éteindre artificiellement » une fraction aléatoire des neurones à chaque étape de l'apprentissage (fraction empiriquement fixée à un quart) est mise en œuvre après chaque couche de *max-pooling* pour améliorer la capacité du réseau à généraliser[8]. La Figure 3 présente cette architecture de façon schématique.

La sortie du CNN est passée à une couche « entièrement connectée » de 40 neurones (toujours avec la fonction d'activation ReLU puis à une dernière couche ayant autant de neurones que le nombre de classes (18). Cette couche fait l'objet d'une d'activation par la fonction *softmax*. Cette fonction permet d'entraîner un classificateur à plusieurs classes. La valeur du neurone i de la couche *softmax* correspond à la probabilité que la séquence d'entrée appartienne à la classe i [8].

4.2.3. Variante avec couche de réseau récurrent

De façon à obtenir une plus grande réactivité lors d'un changement de mode de jeu, nous avons cherché à bénéficier de la capacité d'oubli implémentée dans les RNN à porte. Dans cette variante, la couche intégralement connectée est remplacée par un réseau récurrent constitué d'une simple couche de 40 unités de type GRU.

4.3. Configuration d'apprentissage

Le système est entraîné en minimisant la fonction de perte d'entropie croisée sur le jeu d'apprentissage par un processus de descente de gradient par mini-batch. Les développements sont réalisés en Python dans framework Tensorflow. L'initialisation des paramètres se fait en suivant l'initialisation de Xavier uniforme. La méthode d'optimisation ADAM est combinée avec une décroissance exponentielle du taux d'apprentissage dans différents scénarios[11]. [8]

La méthode dite de l'*arrêt prématuré* est utilisée pour éviter le sur-ajustement du réseau : l'apprentissage est arrêté lorsque la fonction de perte sur le jeu de validation n'a pas diminué après 5 *epochs*[12], ou bien arbitrairement au bout de 80 *epochs*. Le temps d'apprentissage varie sensiblement d'un scénario et d'une base de test à l'autre (entre 3 et 5h avec une carte NVIDIA Titan V).

4.4. Evaluation du classificateur

Nous évaluons dans un premier temps le classificateur sur un sous-ensemble homogène de notre corpus de données. Ainsi, nous excluons consécutivement chacune des 5 bases de l'expérience, et le réseau est entraîné sur 90% des séquences choisies au hasard parmi les 4 autres bases, et testé sur le jeu de validation constitué des 10% restants.

Pour évaluer la capacité du classificateur à généraliser, nous adaptons ensuite la méthodologie *minus-1db* à nos bases de données. Nous testons alors le réseau sur la base qui a été exclue du processus d'apprentissage et de validation précédent. Les taux de précision sur le jeu de test sont moyennés sur diverses trajectoires du taux d'apprentissage.

L'analyse des résultats de cette méthodologie doit prendre en compte l'extrême hétérogénéité des 5 bases en termes de couverture des 18 modes de jeu. En effet, d'une banque de sons à l'autre, pour un mode donné, la tessiture possible est très variable. Le nombre d'échantillons fournis pour une même note en fonction du niveau d'intensité (*velocity layers*) n'est pas le même. Certaines banques proposent plusieurs échantillons selon la corde (I-IV) utilisée pour réaliser la note. Enfin, d'autres fournissent des échantillons spécifiques pour les

[10] *Rectified Linear Unit*, soit la fonction y=max(x,0)

[11] Taux initial $\tau \in \{0.01, 0.03\}$. Amortissement de 0,1% tous les 100 mini-batchs de 128 ou 256 séquences.

[12] Désigne une itération consistant à parcourir une fois en entier l'ensemble des données d'apprentissage avec l'algorithme de descente de gradient.

notes répétées, les notes tenues avec de fortes modulations d'intensité, et même pour l'enchaînement de certains intervalles.

Il arrive donc fréquemment dans nos expériences qu'un mode de jeu donné soit représenté dans une base de test avec une variabilité intra-classe très importante par rapport à celle que le réseau aura rencontré dans les autres bases sur lesquelles il a été entraîné. Dans la méthodologie *minus-1db*, il faut s'attendre alors à ce que la précision du classificateur soit faible pour ce mode et cette base de test.

Enfin, nous n'évaluons pas ici la capacité de généralisation du classificateur construit avec la contribution de *toutes* les bases. Un test complémentaire devra être effectué sur de véritables enregistrements de solos instrumentaux correctement labellisés, mais ceci sort du cadre de cette première étude.

5. RESULTATS

5.1. Corpus homogène

Le classificateur entraîné et testé sur un jeu homogène de 4 bases de données atteint de façon systématique des taux de précision supérieurs à 92%, comme l'indique la table 3.

Dans le cas de ce corpus homogène, on constate une précision supérieure pour le réseau avec la couche de récurrence. Néanmoins le gain en précision est peu significatif statistiquement (Student t-test p-value=27,5%). Ceci s'explique probablement par la capacité très importante mise en œuvre pour le CNN.

Précision	CNN+IC	CNN+RNN
CONT	92,90%	93,10%
EWQL	94,00%	94,15%
ISI	95,20%	95,50%
VSL	97,30%	97,60%
VO	93,40%	93,54%

Table 3. Précision maximale obtenue avec les deux variantes du classificateur entraînées sur un corpus homogène (en ligne, la base exclue du corpus).

5.2. Corpus hétérogène (méthodologie *minus-1db*)

Testé sur la base restante, le réseau obtient des résultats plus contrastés, avec des précisions moyennes autour de 50% pour ISI et CONT et des résultats plus faibles sur les trois autres bases (Table 4).

L'utilisation de la variante avec RNN n'apporte pas d'amélioration uniforme à la précision obtenue. La difficulté à maîtriser la capacité d'oubli du RNN peut expliquer ce résultat.

Précision	CNN+IC	CNN+RNN
CONT	**49,93%** ± 1,06%	49,46% ± 1,38%
EWQL	30,25% ± 0,95%	**32,26%** ± 2,21%
ISI	51,10% ± 0,83%	**51,76%** ± 1,64%
VSL	**43,98%** ± 0,70%	42,58% ± 1,70%
VO	32,42% ± 1,28%	**33,28%** ± 1,28%

Table 4. Moyenne et écart-type de la précision obtenue avec les deux variantes architecturales du classificateur testées sur la base en ordonnée et entraîné sur la réunion des 4 autres.

5.3. Facteurs influençant la précision obtenue

5.3.1. Nombre de bases disponibles

Dans le cadre du modèle CNN+RNN testé sur la base ISI, la table 5 indique le taux de précision moyen[13] en fonction du nombre de bases du jeu d'apprentissage dans lesquelles les MJI sont représentés.

Nombre de bases	Taux de précision
1	35,52%
2	39,18%
3	73,28%
4	70,37%

Table 5. Relation entre taux de précision moyen et nombre de bases dans lesquelles les modes de jeu ont été rencontrés à l'apprentissage (Base de test=ISI, Architecture = CNN+RNN).

Une relation semble se dessiner, selon laquelle la capacité de généralisation croîtrait globalement avec le nombre de bases indépendantes disponibles. Elle doit cependant être vérifiée sur tous les modèles.

5.3.2. Position de l'attaque au moment de la prédiction

L'un des enjeux d'un classificateur en temps réel est de présenter la plus grande réactivité possible à un changement de mode de la part de l'instrumentiste. Pour mesurer cette réactivité, nous nous intéressons uniquement aux séquences où s'opère un changement de mode. Nous calculons la valeur moyenne de la précision du système en fonction du nombre de fenêtres séparant l'attaque de la note et la prédiction effectuée (figure 3).

[13] Taux moyen non pondéré calculé à partir des taux moyens par classe.

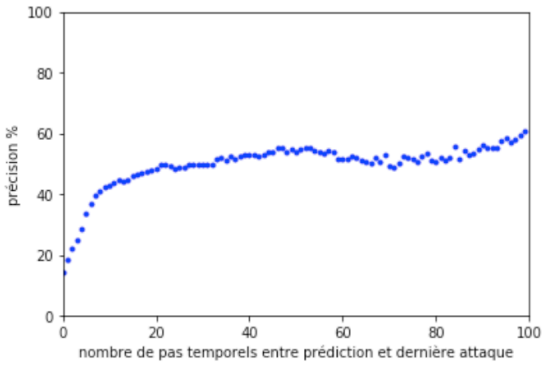

Figure 3. Moyenne de la précision obtenue pour les séquences avec changement de mode en fonction de la distance à l'attaque, tous modes confondus (Base de test=ISI, Architecture = CNN+RNN).

Lorsque le nouveau mode de jeu vient de rentrer dans la séquence analysée, le réseau dispose de peu d'information pour réaliser une prédiction. Au fur et à mesure que le temps s'écoule entre l'attaque de la note correspondant au nouveau mode et la prédiction, la précision du réseau augmente. Au bout de 10 fenêtres (environ 210ms) elle dépasse 40% mais continue à croître jusqu'à ce que l'attaque sorte de la séquence analysée (soit 60 fenêtres).

Le comportement temporel du réseau est très variable d'un mode à un autre. L'attaque percussive d'un pizz. Bartok (figure 4a.) est si particulière qu'elle sera reconnue au bout de 4 fenêtres avec une précision proche de 95%. Mais cette précision va s'effondrer lorsque l'attaque disparaîtra de la séquence analysée et que ne subsisteront que des résonances.

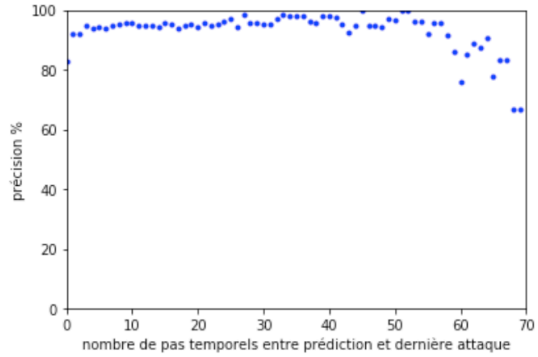

Figure 4a. Cas du mode : *pizz. Bartok*.

Inversement, rien ne permet de distinguer la présence du vibrato dans une note jouée ordinario dans les premiers instants suivant l'attaque, ce qui se traduit par des taux de précision très faibles pour le mode *sustained vibrato* (figure 4b).

S'agissant d'un mode entretenu, le taux de précision croît au fur à mesure que le temps s'écoule après l'attaque.

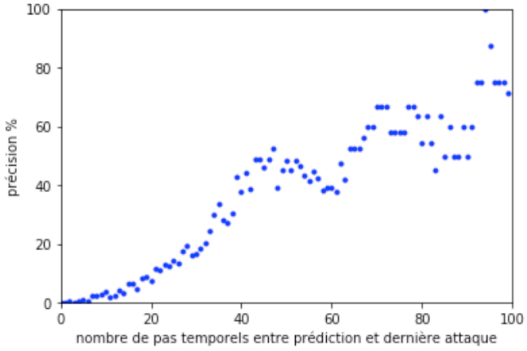

Figure 4b. Cas du mode : *sustained vibrato*.

Ces exemples démontrent l'existence d'un taux d'erreur incompressible de notre système pour les séquences constituées de résonances de modes non entretenus et celles qui sont trop proches d'un changement de mode. Ce type d'erreur n'existe pas dans un système qui n'aurait pas la contrainte du temps réel et ne chercherait à analyser que des notes complètement formées (avec attaque et *release*).

5.4. Matrice de confusion et analyse de cas

La matrice de confusion du modèle présenté dans la section 5.3. (figure 5) fait apparaitre une grande variabilité dans les taux de précision des différents modes, de 97,83% pour le *pizz. Bartok* à seulement 0,2% pour le *tremolo sul tasto*.

Dans ce dernier cas, le caractère *sul tasto* du tremolo n'a pas été identifié par le réseau, qui a classé à 85% les séquences correspondantes comme de simples *tremolos*. Des séquences *sul tasto*, le réseau a retenu surtout (à 29,5%) l'absence de *vibrato*. On mesure ici les limites d'une classification unique pour des phénomènes multi-dimensionnels.

Certains cas prévisibles de confusion liés à l'hétérogénéité des bases méritent d'être signalés. On en prendra deux exemples.

D'abord, le mode *détaché* correspond dans la base ISI à des échantillons de notes courtes non vibrées, enchaînés sans *legato*[14] : il se retrouve largement confondu (à 45%) avec le *sustained non vibrato*.

Ensuite, le mode *staccato/spiccato* est massivement confondu avec le mode Sfz/*marcato*. En-dehors d'ISI, le premier est représenté principalement par les techniques du *spiccato* et du *jeté* alors qu'ISI ne contient que du *staccato*. Inversement, en-dehors d'ISI, la classe Sfz/*marcato* est dominée par des échantillons *marcato* (interaction brève), le sforzando (interaction longue) étant justement une contribution d'ISI. Ces spécificités dans les contributions des bases expliquent un taux d'erreur très élevé.

[14]Sans superposition entre *release* d'une note et attaque de la note suivante.

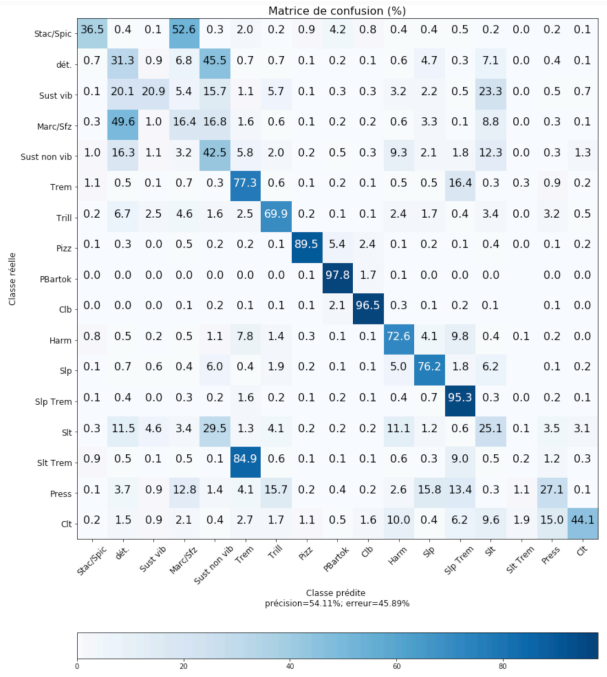

Figure 5. Matrice de confusion d'un modèle avec couche récurrente, base de test ISI. En ligne, les classes réelles, en colonne, les classes prédites par le réseau.

6. EXEMPLES D'APPLICATION

6.1. Suivi de partition

Les systèmes de suivi de partition comme Antescofo [5] sont capables de comparer des signaux audio complexes (monophoniques et polyphoniques) avec une partition symbolique et d'opérer une synchronisation entre les deux. Pour autant, cette comparaison s'arrête au domaine des hauteurs et ne s'étend pas aisément à la question du timbre. Il n'est pas possible de synchroniser le système lorsqu'un violoncelliste frappe le corps de son instrument ou vient jouer derrière le chevalet. A fortiori il sera impossible d'évaluer le tempo de l'interprétation et donc de déclencher un événement en respectant ce tempo.

Ainsi le classificateur de modes de jeu proposé dans cet article pourrait contribuer à régler cette difficulté.

Au lieu de devoir déclencher des évènements manuellement dans pareil cas, leur gestion serait donc intégrée au système de suivi et bénéficierait de la musicalité que permet le lancement de traitements dans le tempo instantané de l'interprétation.

6.2. Systèmes d'improvisation

Les systèmes d'improvisation comme Omax [12] ou Improtek [18] sont dotés d'une capacité d'écoute et d'analyse, qui leur permet de transformer le flux audio émanant du partenaire d'improvisation en un flux symbolique. Les deux flux synchronisés sont stockés dans une base de connaissances que le système vient ensuite interroger pour définir sa réaction aux stimuli externes.

Les possibilités de synthèse et de traitement du signal en temps réel[15] de ces systèmes sont aujourd'hui peu développées. En effet, il faudrait que le système puisse déterminer, parmi les modules de synthèse et de traitement élémentaires[16] qui seraient disponibles, ceux qui sont les plus appropriés compte-tenu du flux audio à transformer. Ainsi, des modes de jeu percussifs seront sans doute plus adaptés à certains types de transformations de nature rythmique comme des lignes à retard alors que des modes de jeu fortement bruités (e.g. cordes écrasées, *con legno tratto*, jeu au-delà du chevalet) ou instables sur le plan fréquentiel (e.g. trilles d'harmoniques) donneront des résultats plus intéressants à la synthèse granulaire qu'une note tenue *ordinario*. A ce jour, Omax et Improtek permettent de segmenter le flux audio selon des critères liés au timbre instrumental. En revanche, cette segmentation s'opère par clustering dans un espace de descripteurs continus[17]. Elle ne permet pas de prendre le type de décisions évoquées ci-dessus, contrairement au classificateur de modes de jeu proposé dans cet article.

Notre système contribuerait donc au développement d'une capacité générative susceptible de s'adapter au partenaire instrumental. Ces routines génératives viendraient s'ajouter aux puissants algorithmes de réinjection stylistique déjà mis en œuvre.

7. CONCLUSION

Dans cet article, nous avons étendu les méthodes de l'état de l'art en matière de reconnaissance du timbre instrumental dans les textures polyphoniques à la classification de modes de jeu en temps réel à partir d'enregistrements de solistes.

Un classificateur construit à partir d'un CNN profond (combiné ou non avec une couche de récurrence) nous permet d'obtenir d'excellents résultats sur un corpus homogène provenant de 5 banques de sons. Cependant, sa performance s'affaiblit sensiblement lorsqu'on met en œuvre la méthodologie *minus-1db*, ce qui pourrait indiquer une faible capacité du modèle à généraliser.

Comme piste de recherche future afin de préciser notre évaluation sur ce point, une nouvelle base de test sera constituée à partir d'un corpus de 8 heures de solos de violoncelle contemporain, partiellement labellisé à

[15] Ou, s'agissant des enregistrements stockés dans la base de connaissance, en temps différé.
[16] Par ex. des lignes à retard, des harmonizers, des bancs de filtres résonnants.
[17] En l'espèce, les MFCC.

partir des partitions des œuvres. Une refonte de la modélisation des classes de MJI sera mise en oeuvre afin de prendre en compte leur multi-dimensionnalité naturelle, en combinaison avec une logique d'apprentissage multi-tâche [2] . Dans le cadre de cette refonte, une version ouverte des bases de données dans le standard JAMS d'annotation [11] sera mise à disposition de la communauté afin de faciliter la reproductibilité des expériences. A des fins de comparaison, nous implémenterons également l'état de l'art dans le domaine de la classification instrumentale à partir de solos (diverses variantes de CNN s'appuyant sur des CQT, permettant théoriquement de mieux capter l'enveloppe spectrale des sons [15]). Afin de faciliter la généralisation du réseau, nous étudierons l'utilisation d'une composante d'apprentissage non supervisé [7] sur la partie non labellisée du corpus de solos.

8. RÉFÉRENCES

[1] Bhalke, D.G., Rama Rao, C.B., et Bormane, D.S., « Automatic musical instrument classification using fractional Fourier transform based-MFCC features and counter propagation neural network ». Journal Int. Inform. Syst. Vol. 46.3, pp. 425–446, 2016.

[2] Caruana, R., « Multitask Learning ». Machine Learning, vol 28, 1997.

[3] Chen, Y.-P., Su, L., et Yang, Y.-H., « Electric Guitar Playing Technique Detection in Real-World Recording Based on F0 Sequence Pattern Recognition ». In Proc. International Society for Music Information Retrieval (ISMIR), 2015.

[4] Choi, K., Fazekas, G., Cho, K., et Sandler M., « A Tutorial on Deep Learning for Music Information Retrieval », arXiv:1709.04396v2, 2018.

[5] Echeveste, J.M., « Un langage de programmation pour composer l'interaction musicale : la gestion du temps et des événements dans Antescofo », Thèse de doctorat de l'université Pierre et Marie Curie, Paris, 2015.

[6] Essid, S. « Classification automatique des signaux audio-fréquences : reconnaissance des instruments de musique. » Thèse de doctorat de l'Université Pierre et Marie Curie, Paris, 2005.

[7] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V., « Domain-Adversarial Training of Neural Networks », Journal of Machine Learning Research, vol. 17, p. 1-35, 2016.

[8] Goodfellow, I., Bengio, Y., et Courville, A., Deep Learning, MIT Press, Cambridge MA, USA, 2017.

[9] Han, Y., Kim, J., Lee, K., « Deep convolutional neural networks for predominant instrument recognition in polyphonic music ». IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 25, no. 1, pp. 208–221, 2017.

[10] Humphrey, E.J., Durand, S., et McFee, B., « Open-MIC 2018: An open dataset for multiple instrument recognition ». In Proc. ISMIR, 2018.

[11] Humphrey, E.J. , Salamon, J., Nieto, O., Forsyth, J., Bittner, R.M., Bello, J.P., « JAMS : a JSON annotated music specification for reproducible MIR research », in Proc. ISMIR, 2014.

[12] Levy, B., « Principes et architectures pour un système interactif et agnostique dédié à l'improvisation musicale », Thèse de doctorat de l'université Pierre et Marie Curie, Paris, 2013.

[13] Livshin, A. « Automatic Musical Instrument Recognition and Related Topics. Acoustics », Thèse de doctorat, Université Pierre et Marie Curie, Paris VI, 2007.

[14] Lostanlen, V., Andén, J. , et Lagrange M.. «Extended playing techniques: the next milestone in musical instrument recognition ». In Proc. DlfM, Paris, France, 2018.

[15] Lostanlen, V., Cella, C.-E., « Deep convolutional networks on the pitch spiral for music instrument recognition », In Proc. ISMIR, 2016.

[16] Loughran, R., Walker, J., O'Neill, M. et O'Farrell, M.,« The Use of Mel-frequency Cepstral Coefficients in Musical Instrument Identification », In International Computer Music Conference (ICMC), 2008.

[17] Loureiro, M.A., Hugo Bastos de Paula, H. et Yehia, H.C., « Timbre Classification Of A Single Musical Instrument ». In Proc. ISMIR, 2004.

[18] Nika, J., Chemillier, J.-M., Assayag, G., « Improtek: introducing scenarios into human-computer music improvisation », ACM Computers in Entertainment (CIE), 2017.

[19] Patil, K., et Elhilali, M., « Biomimetic spectro-temporal features for music instrument recognition in isolated notes and solo phrases. » EURASIP J. Audio Speech Music Process, 2015

[20] Pons, J., Slizovskaia, O. Gong, R., Gómez E. et Serra X., « Timbre Analysis of Music Audio Signals with Convolutional Neural Networks » in 25th European Signal Processing Conference (EUSIPCO), 2017.

[21] Rowe, R., Interactive Music Systems : Machine Listening and Composing, MIT Press, Cambridge MA, USA, 1993.

[22] Tindale, A.R., Kapur, A., Tzanetakis, G. et Fujinaga, I.. Retrieval of percussion gestures using timbre classification techniques », In Proc. ISMIR, 2004.