Escola Tècnica Superior d'Enginyeria ETSE-UV

Programa de Doctorat en Tecnologies de la Informació, Comunicacions i Computació.


Doctoral Thesis

# Signal Processing Techniques for Robust Sound Event Recognition

**Author:**

Irene Martín Morató

**Advisors:**

Francesc J. Ferri

Máximo Cobos

**September, 2019**

Escola Tècnica Superior d'Enginyeria ETSE-UV

Programa de Doctorat en Tecnologies de la Informació, Comunicacions i Computació.


Doctoral Thesis

# Signal Processing Techniques for Robust Sound Event Recognition

**Author:** Irene Martín Morató

**Advisors:** Francesc J. Ferri and Máximo Cobos

**September, 2019**

This work has been partially supported/funded by FEDER and Spanish Government through projects TIN2014-59641-C2-1-P, RTI2018-097045-B-C21, and FPU14/06329.

*Als meus pares i Nuria.*

# Agraïments

Aquesta tesi no hagués estat possible sense tota la gent que m'ha ajudat i m'ha donat suport en tots els sentits. A totes aquestes persones vull dedicar aquesta tesi, i agrair-los tot el que han fet per mi.

Especial menció per als meus directors de tesi, Francesc J. Ferri i Màximo Cobos, per haver-me guiat, inspirat i motivat durant aquest trajecte. Gràcies per confiar en mi des del primer moment i haver-me fet costat en els moments més durs.

També agrair especialment als meus pares, per estar sempre al meu costat i donar-me força quan més la necessitava.

Thanks to all the members of the audio research group at the Tampere University, for welcoming me and for the inspiring and motivating environment of which I was fortunate to be part of. Kiitos paljon!

# Resum

Avanços recents en el camp de l'intel·ligència artificial, han afavorit la incorporació d'aquesta tecnologia al nostre dia a dia. Cada vegada més, els sistemes intel·ligents permeten una fàcil interacció, sent aleshores accessibles per a un usuari mitjà. Aquesta tecnologia està molt relacionada amb l'augment en popularitat de sistemes basats en la Internet de les coses (Internet of Things, IoT), la qual cosa està donant lloc al desenvolupament de noves aplicacions amb la possibilitat d'utilitzar-les en qualssevol dispositiu. Aquestes aplicacions, possibiliten la realització de tasques quotidianes dutes a terme per màquines intel·ligents amb resultats semblants als obtinguts per una persona. Exemples d'aquestes tasques són, el reconeixement facial, la transcripció de textos, la classificació automàtica de gèneres musicals o el reconeixement d'entorns acústics. Dintre del domini acústic, camps relacionats amb l'estudi de la parla o l'anàlisis musical conformen una extensa branca d'investigació i desenvolupament que ha possibilitat aplicacions com Alexa (de Amazon) o Google assistant, així com el sistema avançat de recomanacions musicals ofert per Spotify. També recentment, tasques relacionades amb l'anàlisi d'entorns acústics estan atraient més investigadors degut a l'àmplia possibilitat d'aplicacions relacionades.

Comprendre i analitzar el nostre entorn a partir de la percepció del so, és una habilitat utilitzada des de sempre, tant per animals com per humans, per a poder sobreviure en l'entorn i poder interactuar amb ell. No obstant això, aquesta capacitat auditiva es veu afectada en algunes persones. Al voltant de 466 milions de persones arreu del món pateixen algun grau de pèrdua auditiva, i s'estima que aquesta quantitat augmente fins als 900 milions en 2050. Sols en Europa, 52 milions de persones, un 10% de la població total, té deficiència auditiva. L'ús de sistemes intel·ligents capaços d'imitar l'oïda humana millorarien considerablement la qualitat de vida d'aquestes persones. D'altra banda, hi ha situacions en què no és possible la presència humana ni la instal·lació de càmeres de video per a monitoritzar l'entorn. Exemples d'aquestes situacions són la vigilància de vivendes per qüestions de privacitat o llocs de difícil accés, on l'àrea a cobrir és massa gran per a monitoritzar-la. Aquestes situacions, han motivat l'ús de sensors acústics capaços de realitzar tasques relacionades amb l'anàlisi i reconeixement de senyals acústics. El desenvolupament de màquines amb la destresa de processar, analitzar i reconèixer els senyals capturats pels sensors acústics permet d'implementar aplicacions capaces de resoldre els problemes mencionats prèviament. A l'hora de fer front a tasques relacionades amb el processament d'entorns acústics, és convenient introduir el concepte d'event sonor/acústic/auditiu (*sound or audio event* en anglès) com un so específic produït per una font sonora. I també el concepte d'escena sonora/acústica/auditiva (*acoustic or audio scene* en anglès) com un conjunt d'events en un determinat entorn percebut o experimentat per un determinat receptor.

Actualment existeixen aplicacions encarregades de monitoritzar un conjunt controlat d'events sonors que tenen lloc en un entorn tancat. Aquestes permeten notificar en temps real a l'usuari, mijantçant canvis de llums o alertes en dispositius personals intel·ligents. També hi ha treballs de recerca amb l'objectiu de monitoritzar la població aviària a la natura, altres estan més enfocats a l'estudi de la contaminació acústica existent en les grans ciutats

i que ocasiona problemes de salut en la població. Per a resoldre aquestes tasques, es poden utilitzar diferents mètodes de reconeixement de patrons acústics. Depenent del tipus de dades d'entrenament de què disposem, aquests mètodes es poden diferenciar en supervisats (les dades d'entrenament disposen de les seves corresponents etiquetes), no-supervisats (sols disposem de les dades d'entrenament, es tracta d'agrupar mijantçant característiques, els diferents conjunts de dades) o semi-supervisats (on únicament disposem d'etiquetes d'un conjunt reduït de la base de dades). Els sistemes de reconeixement que aconsegueixen un rendiment major són els entrenats de forma supervisada. Aquesta tècnica implica l'obtenció d'una quantitat considerable d'exemples acústics representatius del conjunt de dades que conformen la tasca en qüestió. Al ser un entrenament supervisat les dades recollides han d'estar degudament etiquetades, indicant la classe a la que pertanyen i els temps de començament i acabament del event sonor. Aquest tipus d'anotació dóna lloc al que es coneix com dades fortament etiquetades (*strong labels*). Gràcies a la reducció de cost i a l'augment de capacitat dels dispositius electrònics actuals, és possible obtindre grans quantitats de dades, enregistrades en diferents escenaris i situacions. La dificultat ve quan s'han d'anotar les dades recollides de les quals s'ha d'identificar individualment cada un dels events que formen l'escena acústica. Aquesta tasca sol ser realitzada per experts anotadors que escolten minuciosament els senyals acústics per poder fixar el nom de la classe a què pertanyen i en quin interval temporal es produeixen. Per intentar evitar errors d'interpretació, més d'un expert analitza les dades i l'anotació resultant és un consens entre les opinions dels experts. Aquest procés resulta ser el més costós en termes de temps i recursos quan es vol dissenyar un sistema de classificació. A més, la perfecta realització de la fase d'anotació no implica un perfecte entrenament, ja que l'anotació és duta a terme per humans, el que implica que sempre existirà un factor de subjectivitat inherent el qual hi haurà que tractar de mitigar amb diferents tècniques. El conjunt de dades perfectament etiquetades és l'utilitzat per a entrenar, el que es coneix com dades d'entrenament, les quals, com hem mencionat, han sigut acuradament processades. En canvi, les dades reals, utilitzades una vegada la màquina està entrenada, no solen tindre les mateixes condicions acústiques, on el soroll de fons o el nivell de solapament entre diferents events, que pertanyen a la mateixa o distinta classe, pot variar.

## Objectius

Aquesta tesis s'emmarca en el camp concret de reconeixement d'events sonors, on l'objectiu es identificar els events sonors i assignar-los la classe corresponent per a poder interpretar-los. El sistema auditiu humà és molt bo per reconèixer la complexa barreja de sons present al nostre entorn. És capaç d'identificar si el so d'interès prové de diverses fonts (que es poden fixar o moure) o diferenciar les característiques espectrals de la senyal acústica, que poden ser tonals (alarmes) o semblants al soroll (xarrades, multituds). A més, es poden classificar els sons en funció del seu comportament temporal, transitori (com per exemple, explosions), continu (dividit en sons estacionaris, com el soroll d'un motor i no estacionari, per exemple, parla humana) o sons intermitents amb trets periòdics (soroll de passos) o amb intervals irregulars (lladruc de gos). Per tant, un sistema d'aprenentatge dirigit a interactuar amb el nostre entorn d'una manera similar a com ho fan els humans, ha de ser capaç de reconèixer la gran varietat de senyals acústiques. A més, ha de poder generalitzar quan les condicions acústiques de les dades d'entrenament són distintes a les de test. Aquestes condicions engloben situacions típiques que poden ocórrer en entorns tancats o oberts, com és la reverberació, la interferència d'altres events sonors o la presència de soroll de fons. També pot aparèixer soroll degut a interferències ocasionades pels dispositius d'enregistrament, que poden variar respecte als utilitzats durant l'adquisició dels events d'entrenament.

L'objectiu principal de la tesi és millorar la robustesa dels sistemes de classificació quan hi ha situacions acústiques adverses diferents a les de l'entrenament, com és la presència de

soroll de fons o quan els events no han pogut ser perfectament emmarcats temporalment i existeix per tant soroll de segmentació.

Aquest objectiu principal es pot desglossar en sub-objectius els quals adrecen problemes més concrets amb més detall. Un dels objectius de la tesi és estudiar el problema de classificar events sonors i com es diferencia d'altres tasques relacionades amb l'estudi del so. També és donar una visió general del camp de l'anàlisi de sons ambientals i quin es l'estat actual de la tècnica, remarcant les aplicacions existents que es veuen beneficiades per aquest camp. Una vegada s'han introduït les diferents metodologies supervisades existents en la literatura, es decanta per l'ús de màquines de vectors suport (Support Vector Machines, SVM) com a model per realitzar la classificació, motivat principalment per l'escassesa de dades acústiques fortament etiquetades. Abans de donar com a entrada al model de classificació els senyals acústics, aquests són analitzats, extraient les característiques més representatives per tal de facilitar la feina de classificació. Les principals característiques utilitzades per a representar events sonors, venen motivades pels seus bons resultats en altres tasques pertanyents al domini de l'àudio, com son les tasques de reconeixement de veu o classificació de gèneres musicals. Exemples d'aquestes característiques són: coeficients cepstrals en les freqüències de Mel (Mel-Frequency Cepstral Coefficients, MFCCs), àmpliament utilitzats en el reconeixement automàtic de la parla, o la energia en bandes de Mel (Mel-Frequency Energy, MEL), les quals han guanyat popularitat per a problemes relacionats amb events sonors. En aquest context, un dels principals objectius és l'estudi en termes de sensibilitat i robustesa d'aquestes propietats. Aquest estudi està enfocat en ajudar a millorar el procés d'extracció de característiques. Per tant, en aquesta tesi, es compara com afecten diferents conjunts de característiques al rendiment final del classificador.

L'aparició de noves tècniques d'entrenament supervisat basades en xarxes neuronals profundes (Deep Learning Networks, DNN) enfoca de forma diferent el pas d'extracció de característiques. Mijantçant DNN, i amb suficients dades d'entrenament, el sistema pot aprendre noves representacions mijantçant les activacions internes del model. Tant aquestes representacions internes (conegudes com deep features) com les característiques més clàssiques (hand-crafted), són utilitzades per a entrenar un model de classificació obtenint resultats comparables als de l'estat de l'art. Però quan les condicions acústiques de les dades de test no son òptimes, és a dir, les propietats acústiques són diferents de les dades d'entrenament, les característiques no són capaces de representar apropiadament les propietats distintives dels events per a poder diferenciar-los. Aquesta situació motiva el disseny de tècniques de processat per a millorar la robustesa d'aquestes característiques quan hi ha condicions acústiques adverses, definint un altre objectiu important de la tesi. Un nou mètode de processat aplicat una vegada les característiques s'han calculat, permet fer més robust el sistema davant errors de segmentació. Aquest mètode es basa en capturar l'evolució temporal de l'event sonor per a identificar millor les parts rellevants del senyal. Finalment, la millora de les característiques més clàssiques en aplicar aquest nou mètode, motiva el disseny d'un nou model que combine les deep features amb tècniques adaptatives per a capturar la informació rellevant representada per les característiques. Els coneixements adquirits durant l'estudi de l'energia com a ferramenta de representació dels canvis del so al llarg del temps, motiven l'últim objectiu de la tesi. Es tracta d'aplicar una nova representació que permet ser aplicada a problemes d'anotació on la subjectivitat humana presenta un problema. Per últim, mijantçant models que permeten ser entrenats utilitzant aquesta informació es pot reduir l'efecte perjudicial d'aquesta subjectivitat.

## Metodologia

Per a dur a terme aquesta tesi, primer s'ha fet un extens estudi de l'estat de l'art en el camp de la classificació i detecció d'events sonors. La principal diferència entre ambdues tasques,

és que en els problemes de classificació, únicament la classe a què pertany l'event és d'interès, mentre que en problemes de detecció es requereix a més la identificació dels temps d'inici i final de l'event, la qual cosa fa que el problema de detecció siga més complex i per tan es prefereix l'estudi de tècniques inspirades en tasques de classificació que puguen ser fàcilment aplicades a problemes de detecció.

Per a poder mesurar el rendiment del model de classificació s'han de definir mètriques d'avaluació. No hi ha consens sobre quina mètrica és millor per avaluar el rendiment global d'un sistema de classificació d'events sonors. Mètriques com la precisió, la freqüència d'error d'esdeveniments acústics (acoustic event error rate, AEER) o l'àrea sota la corba (area under the curve, AUC) han estat àmpliament adoptades per la comunitat d'àudio com a estàndard a efectes de comparació i avaluació de diferents models utilitzant bases de dades distintes. Al llarg de la tesi, s'ha utilitzat com a model d'aprenentatge principal un sistema supervisat basat en SVMs. Gràcies al reduït nombre de paràmetres necessaris per a entrenar aquests models, s'ha realitzat un estudi minuciós de la importància de les característiques clàssiques en el problema de classificació. Per a realitzar aquest estudi, s'ha utilitzat un algoritme de selecció de característiques iteratiu basat en dos criteris d'avaluació: la probabilitat a posteriori del model i l'encert. Amb aquest algoritme s'ha obtingut l'ordre d'importància de les característiques MFCC, MEL i les seves corresponents deltes, les quals aporten informació sobre la dinàmica temporal de les característiques. Els resultats obtinguts han de ser extrapolats tenint en compte el context, ja que s'ha utilitzat una base de dades molt específica, formada per events sonors pertinents a un entorn d'oficina concret. Exemples de les classes utilitzades son: telèfon, escriure en teclat, colp de claus, porta tancant-se, etc. Així i tot queda motivat l'ús de característiques basades en energia (MFE) per a futures aplicacions, d'acord amb els resultats de l'ordenament per importància de les característiques.

El pas d'extracció de característiques és molt important i ha de ser tractat amb cura ja que la forma en què es representen els arxius d'àudio afecta directament al rendiment del classificador. Els paràmetres relacionats amb l'extracció de les característiques així com el tipus de característiques o el conjunt de característiques seleccionat per a la tasca en qüestió, varien en funció de l'experiència de l'investigador. Si l'investigador té prou coneixement de la tasca, pot aproximar-se a un conjunt de característiques adequat basat en experiments anteriors. No obstant això, fins i tot amb experiència prèvia, les condicions del conjunt de dades són molt canviants, el que pot augmentar la dificultat del problema, resultant en una selecció de característiques inútils. Tradicionalment, aquestes característiques hand-crafted s'han dividit en el domini de temps i domini de freqüència, variant la seva popularitat en funció de l'aplicació final. Recentment, donades les millores en la generalització realitzada per models de xarxes profundes, ha donat lloc a estudiar tècniques per obtindre sense necessitat d'una costosa anàlisi manual, una bona representació dels senyals acústics.

La publicació en obert de l'arquitectura de xarxes neuronals complexes amb els seus paràmetres ja entrenats, permet l'ús de transferència de coneixement (transfer learning). Aquestes xarxes estan entrenades utilitzant grans quantitats de dades fortament etiquetades, en dominis com l'anàlisi d'imatges on el camp de recerca és més antic i la col·laboració és més abundant la qual cosa permet l'aparició de bases de dades més extenses. La capacitat de generalització d'aquestes xarxes permet extreure informació rellevant, tenint com a entrada únicament l'espectrograma de l'event sonor. S'utilitza un d'aquests models de xarxes neuronals, basats en convolucions (convolutional neural network, CNN) com a extractor de característiques, combinat amb un model lineal basat en SVM encarregat de realitzar la tasca de classificació. Finalment, es realitza un estudi de robustesa contra condicions acústiques adverses, com el soroll i la reverberació. Aquest estudi es realitza a través de la comparació de les deep features amb les característiques clàssiques, veient com ambdues queden afectades degut a les condicions mencionades prèviament.

Els entorns acústics de la vida real, no sols es veuen afectats per condicions adverses com

soroll de fons o canvis d'intensitat. Els events sonors que formen els entorns acústics pertanyen a una gran varietat de classes i solen aparèixer superposats. Aquestes situacions no solen ser un impediment per a la persona que està escoltant. En canvi, per a una màquina, el fet de separar events sonors que tenen lloc al mateix temps, és una tasca extremadament complexa. Un primer pas per ajudar en el procés és dividir el senyal en breus extractes per a aïllar els events sonors d'interès. Hi ha diverses tècniques per a dur a terme aquesta divisió, com és l'ús d'algoritmes de detecció d'inici de senyal o d'altres que permeten trobar el llindar d'energia. Aquestes tècniques es poden agrupar en mètodes basats en segmentació, on abans d'utilitzar el senyal per a classificar, ha de ser processat per identificar on es troba l'esdeveniment d'interès. Existeixen altres tècniques, conegudes com seqüencials, les quals no necessiten aquest preprocés, on el senyal és analitzat utilitzant finestres de longitud fixa que es van desplaçant pel senyal fins recórrer-lo completament. El desavantatge d'aquestes últimes tècniques, és que el model ha d'estar prèviament entrenat per a poder descartar els segments que continguen únicament soroll. En canvi, amb els mètodes basats en segmentació, és possible eliminar prèviament les parts que no aporten informació. Cal tindre en compte que la segmentació difícilment és perfecta, per tant poden quedar parts amb soroll solapant les parts que contenen l'event. L'evolució temporal dels esdeveniments sonors és una informació molt representativa que pot ser molt útil per a reconèixer els sons ambientals. Un dels problemes a resoldre quan es tracta d'analitzar events sonors, és la variabilitat existent en la seua duració. Fins i tot pertanyent a la mateixa classe, degut a la naturalesa d'alguns sons, la duració pot variar d'uns pocs mil·lisegons a uns quants segons. Degut a limitacions computacionals, els models d'aprenentatge requereixen d'una entrada de grandària fixa, la qual cosa suposa allargar la longitud dels events més curts (be afegint ceros o be replicant el senyal), o acurtar els que són més llargs. Un alternativa per a processar i tenir en compte aquesta variabilitat en longitud, és extraure informació estadística com la mitjana o la desviació estàndard. Aquesta informació s'extrau de les característiques del senyal prèviament calculades i una vegada fixades les regions i les seues longituds on es volen extraure els estadístics. Utilitzant la informació sobre l'evolució temporal de l'event, és possible adaptar la longitud de les finestres a una longitud variable que dependrà d'on es trobe la informació més rellevant de l'event. Gràcies a l'ús d'aquesta informació s'ha dissenyat una nova tècnica de processat de característiques adaptatives que permet reduir considerablement les parts amb soroll, abans de donar-li-la com a entrada al classificador.

L'efectivitat d'un model acústic sovint es veu obstaculitzada per l'accés limitat a una gran base de dades fortament etiquetada i que cobreix una mostra representativa de la variabilitat present en les dades acústiques. Els models de classificació entrenats utilitzant bases de dades reduïdes i insuficientment variades, poden estar sobre-entrenats i no generalitzar bé a exemples no vistos. Utilitzant tècniques d'augment de dades és possible mitigar el problema però, en augmentar les dades d'entrenament, el procés d'aprenentatge es torna més complex i requereix més temps. És per això, que s'utilitza l'habilitat de generalització de la qual disposen models basats en DNN. En particular, són els models CNN els que mijantçant un entrenament amb extenses bases de dades, ha demostrat tindre una gran capacitat d'adaptació a condicions canviants. En aquesta tesi, es combina la capacitat de generalització d'aquests models amb l'estudi de l'energia dels senyals com a representació de la evolució temporal dels corresponents events sonors. En particular es proposa una nova capa d'agrupació adaptativa (*adaptive pooling*) capaç de realitzar una transformació temporal no lineal per millorar la robustesa del model davant d'errors de segmentació. La corresponent arquitectura millorada ha demostrat ser robusta quan hi ha errors de segmentació al principi, al final o ambdós en els events sonors, quan s'entrena amb mostres perfectament aïllades.

La representació de l'evolució temporal a partir d'un esdeveniment sonor utilitzant la seua envoltant energètica, ha demostrat ser útil per capturar les parts representatives del senyal, el que ha inspirat l'ús d'aquesta informació per alleujar l'efecte perjudicial de la

subjectivitat inherent a l'anotació d'event sonors. Tot i que la quantitat de bases de dades acústiques ha augmentat, segueix sent molt car anotar correctament els temps d'inici i de finalització per a cada event sonor que forma l'escena acústica per a obtindre les anomenades etiquetes *fortes*. Per reduir l'esforç en el procés d'anotació, recentment s'ha introduït l'ús de dades pobrament etiquetades o etiquetes *febles* per tasques relacionades amb la detecció d'events sonors. En el cas d'aquest etiquetatge feble, només es considera la informació sobre la presència o absència d'un event i no s'emmagatzema informació sobre l'interval de temps concret en què es produeix. Recollir aquest tipus d'informació és considerablement més barat i ràpid. No obstant això, l'ús de bases de dades pobrament etiquetades no proporciona el mateix nivell de precisió que un sistema entrenat amb dades fortament etiquetes. Un altre mètode per obtindre de forma ràpida i barata un etiquetatge fort, es combinar sintèticament events manualment aïllats per a formar mescles simulades d'escenes acústiques. Utilitzant aquestes bases de dades és possible avaluar el model de detecció amb precisió minuciosa, ja que les anotacions estan artificialment controlades. L'última aportació de la tesi, consisteix a estudiar com l'ús d'etiquetes basades en informació sobre l'energia de l'event, permeten l'estimació de la envoltant de mescles acústiques, i la posterior detecció dels events que la formen. Per dur a terme aquesta tasca, es defineix un nou tipus d'etiquetatge basat en la combinació de les etiquetes fortes amb l'energia de la mescla sonora. Aquest tipus d'etiquetes són conegudes com etiquetes suaus (*soft labels*). L'objectiu d'entrenar models per a la detecció d'events basats en etiquetes suaus, és estudiar la possibilitat d'entrenar models amb informació extreta únicament de les mescles, sense necessitat de disposar d'anotadors humans.

## Conclusions

El processament de senyals acústics és un tema ampli i desafiant, amb una branca de recerca creixent, on cada dia es desenvolupen noves aplicacions. Per tant, si volem que el sistema aprenga la informació rellevant dels senyals d'àudio, és a dir, trobar semblances i diferències entre les dades acústiques, és millor si l'entrenem amb una quantitat considerable de dades. Tot i això, no sempre és possible confiar en les dades d'entrenament disponibles. Aquestes dades poden estar danyades o poden contindre soroll, perjudicant la informació rellevant sobre la o les classes d'interès. El primer pas per a dissenyar un sistema de classificació és recopilar dades d'entrenament fiables i ha de ser una mostra representativa del conjunt de dades que formen el problema a resoldre. En segon lloc, la manera de presentar les dades al classificador també és important, on l'elecció de la representació ve donada per la tasca de classificació de què es tracte. Aquesta representació es basa en seleccionar un conjunt de característiques que extreuen informació representativa de les dades i redueixen la seva dimensionalitat. La selecció de les característiques afecta directament al rendiment de la tasca de reconeixement, per tant, aquest procés s'ha d'estudiar amb deteniment.

En aquesta tesi s'han estudiat els diferents conjunts de característiques i la seva rellevància per a la classificació d'events sonors. No obstant això, durant els darrers anys, la popularitat de les xarxes neuronals, que utilitzen estructures lògiques inspirades en el funcionament del cervell humà, ha posat en dubte la importància d'aquest pas d'extracció de característiques. L'ús d'aquestes xarxes com a extractors de característiques es basa en la capacitat de generalització de què disposen, gràcies a la disponibilitat de grans quantitats de dades d'entrenament. El problema de les grans bases de dades és el procés d'etiquetar (assignar una classe a un fragment d'àudio) que requereix molt de temps i és costós. Per fer front a aquestes bases de dades, ha sorgit el concepte d'aprenentatge de transferència, que consisteix a utilitzar els coneixements apresos d'un domini (per exemple, imatges) i transferir-lo a un altre (per exemple, àudio). En aquesta tesi s'han realitzat experiments que utilitzen aquesta metodologia, atenent a la capacitat de generalització d'aquestes característiques en condicions adverses.

És tan important seleccionar un conjunt representatiu de característiques (ja siguen seleccionades manualment o utilitzant tècniques relacionades amb xarxes neuronals), com la capacitat de generalitzar un exemple desconegut. No és suficient implementar un sistema de classificació que funcione bé quan les dades reals de l'aplicació tinguen condicions similars o idèntiques a les dades utilitzades en l'entrenament. Si el rendiment es degrada considerablement quan les condicions de test són adverses (es produeixen sorolls de fons no desitjables, els events sonors es tallen o contenen interferències pertanyents a events sonors veïns, entre altres), vol dir que el nostre sistema no és capaç de generalitzar i no és possible aplicar-lo en condicions de la vida real. En aquesta tesi, ens preocupa la gran variabilitat de les condicions presents en els sons acústics, per la qual cosa s'ha dut a terme un estudi en profunditat amb diferents metodologies per millorar la precisió del classificador quan apareixen errors de segmentació i soroll en les dades de test. A més, l'ús de característiques que permeten l'anàlisi de l'evolució temporal dels events sonors, com l'energia del senyal, ha permès una representació més precisa de les característiques rellevants per a la classificació d'aquests events. Aquest mètode, ha sigut utilitzat al llarg de la tesi per a millorar la robustesa de les característiques acústiques, tant les clàssiques hand-crafted com les relacionades amb les xarxes neuronals, les deep features.

**Paraules clau**
*Classificació d'events sonors, màquines de vectors suport, selecció de característiques, xarxes neuronals profundes.*

# Abstract

The computational analysis of acoustic scenes is today a topic of major interest, with a growing community focused on designing machines capable of identifying and understanding the sounds produced in our environment, similar to how humans perform this task. Although these domains have not reached the industrial popularity of other related audio domains, such as speech recognition or music analysis, applications designed to identify the occurrence of sounds in a given scenario are rapidly increasing. These applications are usually limited to a set of sound classes, which must be defined beforehand. In order to train sound classification models, representative sets of sound events are recorded and used as training data. However, the acoustic conditions present during the collection of training examples may not coincide with the conditions during application testing. Background noise, overlapping sound events or weakly segmented data, among others, may substantially affect audio data, lowering the actual performance of the learned models. To avoid such situations, machine learning systems have to be designed with the ability to generalize to data collected under conditions different from the ones seen during training.

Traditionally, the techniques used to carry out tasks related to the computational understanding of sound events have been inspired by similar domains such as music or speech, so the features selected to represent acoustic events come from those specific domains. Most of the contributions of this thesis are based on how such features are suitably applied for sound event recognition, proposing specific methods to adapt the features extracted both within classical recognition approaches and modern end-to-end convolutional neural networks. The objective of this thesis is therefore to develop novel signal processing techniques aimed at increasing the robustness of the features representing acoustic events to adverse conditions affecting the mismatch between the training and test conditions in model learning. To achieve such objective, we start first by analyzing the importance of classical feature sets such as Mel-frequency cepstral coefficients (MFCCs) or the energies extracted from log-mel filterbanks, analyzing as well the impact of noise, reverberveration or segmentation errors in diverse scenarios. We show that the performance of both classical and deep learning-based approaches is severely affected by these factors and we propose novel signal processing techniques designed to improve their robustness by means of the non-linear transformation of feature vectors along the temporal axis. Such transformation is based on the so called event trace, which can be interpreted as an indicator of the temporal activity of the event within the feature space. Finally, we propose the use of the energy envelope as a target for event detection, which implies the change from a classification-based approach to a regression-oriented one.

**Keywords**
*Sound event recognition, audio classification, deep learning, support vector machines, feature selection, convolutional neural networks.*

# Contents

# List of Figures

# List of Tables

# Acronyms

| | |
|---|---|
| AEER | Acoustic Event Error Rate |
| ASC | Acoustic Scene Classification |
| BPTT | Back Propagation Through Time |
| CASA | Computational Auditory Scene Analysis |
| CNN | Convolutional Neural Network |
| CRNN | Convolutional-Recurrent Neural Network |
| DNN | Deep Neural Network |
| DCASE | Detection and Classification of Acoustic Scenes and Events |
| EBR | Event to Background Ratio |
| ESC | Environmental Sound Classification |
| GAP | Global Average Pooling |
| GRU | Gated Recurrent Units |
| LSTM | Long Short-Term Memory |
| MFCC | Mel Frequency Ceptral Coefficient |
| MFE | Mel Frequency Energy |
| MIR | Music Information Retrieval |
| MSE | Mean Squared Error |
| RNN | Recurrent Neural Network |
| ROC | Receiver Operating Characteristics |
| SBP | Sensitivity Based Prunning |
| SEC | Sound Event Classification |
| SED | Sound Event Detection |
| SER | Sound Event Recognition |
| SNR | Signal to Noise Ratio |
| SVM | Support Vector Machine |

# Chapter 1

# Introduction

Over the past few years, the number of connected devices worldwide has increased considerably and continues to grow each year. A wide range of electronic devices such as sensors, smartphones, smartwatches, laptops, etc. are connected 24 hours bringing together what we know as the Internet of Things (IoT). These devices share tons of data in an easy way, and have many different applications, since they are not only meant for personal use but they are also a key tool for work. Nowadays, it is difficult to imagine spending a day without using them. Governments have seen the potential of IoT platforms and are investing in the development of IoT-related applications. As part of the European programme "Horizon 2020", in 2016 the IoT European Platform Initiative (IoT-EPI) was launched, which is aimed at merging the physical and virtual world creating smart environments. The platform is compound by 7 projects each of them focusing on different levels of abstraction, from hardware capabilities up to API design, aimed at building a robust IoT infrastructure. On the other hand, these interoperability allows companies to collect large quantities of data from multiple locations, devices and users. Thereupon, favoring the emergence of a wide range of smart applications. For example, we can save energy in the city by monitoring the use of public buildings. More concretely, by knowing the amount of people present in one space the energy consumption of the heating system can be reduced. Another example could be reducing the pollution by monitoring the traffic and creating alternative paths. The application can alert users in real time and advice them to avoid certain routes, which can also help to reduce the noise in the city. With regard to health care, monitoring elderly people who live alone will be one example. These devices can help them to be connected all day to a specialist that can assist them if necessary.

When we think about monitoring, we usually think about video cameras, using some kind of visual sensor to *see* the surroundings. However, most people are reticent when cameras have to be installed in their own homes, even more if there is the need for some external processing. Another problem with cameras is that they may not be reliable if the light conditions are poor or there is some obstacle between them and the subject of interest. Luckily, there are alternatives, like acoustic sensors, which can work in scenarios where there is no light or with no line of sight. Other advantages are energy consumption and economic costs, which are much lower than those of a camera. As a result, the above application scenarios offer great possibilities for machine listening systems.

Hearing is one of the five senses that have been used by humans and animals as a survival skill during ages. Nowadays, it is not necessary for humans to have this sense as developed as before. However, we still use it to understand our surroundings and communicate with our

environment. Despite its importance in human life, around 5% of the world's population suffers from hearing loss (466 million people). Having machines able to analyze and understand the acoustic signals present in our surroundings will help these people improve the quality of their every-day life. In order to understand the different sounds present in our environment, a machine has to be able to differentiate a huge variety of acoustic signals, such as music, bird signing, speech, car, siren, dog bark, etc. In fact, the human auditory system is really good at recognizing complex mixtures of sounds. It is able to identify if the sounds of interest come from multiple sources (that can be fixed or moving) or differentiate the spectral characteristics of the acoustic signal, that could be tonal (e.g. alarms) or noise-like (e.g. chatter, crowd). Furthermore, sounds can be classified according to their temporal behaviour into transient (e.g. explosions), continuous (divided into stationary sounds, e.g. engine noise, and non-stationary, e.g. human speech or bird song) or intermittent sounds with periodic patterns (e.g. foot steps) or with irregular intervals (e.g. dog bark). Therefore, a machine system aimed at interacting with our environment in a human-like fashion has to be able to recognize a wide variety of acoustic signals.

The field of machine learning is aimed at developing systems capable of learning from data. They recognize patterns and make decisions with minimum human intervention. These algorithms create a mathematical model based on observed data in order to perform a specific task. These algorithms can be used to build systems capable of recognizing the acoustic environment. Fortunately, nowadays is relatively easy to collect the necessary data to train a machine thanks to the internet. Around 1.8 billion of digital images are uploaded every day, positioning visual data as the most shared social media content. Audio content is less popular, but still, YouTube reports that 100 hours of videos are uploaded to their platform each minute. In 2017, Cisco stated that mobile video traffic is 59% of the total data traffic, expected to increase up to 80% in 2020. For more specific audio content, web pages such as freesound[1] reports to have 697 hours of new sounds uploaded during 2018. Thanks to the availability of such multimedia content it is now possible to select specific data to create big collections of audio content suitable for a given task. Nonetheless, audio signal processing is a broad and challenging topic, with a growing research branch that is smoothly merging with machine learning to create powerful applications making out of sound a primary interaction modality.

## 1.1 Motivation

Nowadays, there are many applications using audio as a main source of information. Big companies such as Google or Amazon are investing in personalized home assistant systems relying on voice interaction. These systems work by recognizing speech signals, which is the most popular acoustic signal studied by the research community. It is followed by music, where machines have learnt to distinguish songs and they can even recommend similar ones, like Spotify does. But there is a wider range of acoustic signals besides music or speech. Basically, we are listening to hundred of different acoustic sources everyday. For the vast majority of people, the acts of listening and understanding these sounds are performed seamlessly and straightforwardly. However, for people with hearing difficulties, these tasks can be a real challenge. Environmental sound event processing can improve the living standards of people suffering from hearing-loss by interpreting the everyday acoustic soundscape. The idea is to implement devices capable of identifying the sound events of interest, recognizing them and notifying them to the user in real-time. It is even hoped that for those who have lost their hearing ability these systems will become their ears.

[1] `https://freesound.org/`

For a machine, recognizing the wide variety of every-day sounds is a challenging task. Many advances have been achieved since the computational acoustic scene analysis (CASA) [160] field became a hot research topic within the audio community. Despite the achievements accomplished in the domain, there are still many challenges to overcome when working with environmental acoustic signals. One of the major problems is the need for a large amount of audio data to train machine learning systems to solve a specific task, more acute if deep learning algorithms are used. In the scenario of having enough training data, the model can have trouble recognizing test sound events (unseen during the training step). This situation is due to the mismatch between the train and test data, when the test data has acoustic conditions different from the ones of the training examples. Background noise, different recording devices or even sounds produced by distinct sources but categorized with the same label (clock alarm, phone alarm, fire alarm they are all alarms but from different sources), can cause problems when the machine is trying to recognize sounds.

Given the above context, a reliable data acquisition process is important for a solid training phase. However, this is not always feasible due to many adverse effects occurring in real-world applications, such as the existence of high background noise levels or reverberation. Additionally, some of the signals of interest may be too weak or even partially cut. In this context, the main objective of the present work is to contribute to the improvement of sound recognition systems in such adverse scenarios. This goal includes enhancing the performance of recognition systems trained with few amounts of data by proposing signal processing techniques aimed at improving the robustness of the features used to represent acoustic events, especially in those situations where the test conditions differ substantially from the ones seen during system training. In addition, it is also interesting to look further into novel models capable of addressing common problems arising during the acquisition of new training data, such as the use of binary (hard) labels to describe the activity of natural sound events or the exploitation of weakly annotated datasets.

## 1.2 Objectives

Taking into account the points described above, the main objective of this thesis can be summarized as follows:

*To design, implement and evaluate signal processing strategies aimed at improving the robustness of sound event recognition systems in adverse acoustic scenarios, i.e. those where the background and segmentation noise characteristics of test data differ substantially from those of the examples used in the training phase.*

This main objective can be broken down into the following sub-objectives:

- To provide an overview of the sound event recognition research field, including its origins, the current state-of-the-art and the emerging applications that are benefiting from this field.
- To analyze common audio event representation features in terms of sensitivity and robustness to multiple degradations. In this sense, we consider both classical hand-crafted features used in traditional recognition approaches and deep features learned by the inner layers of deep neural networks.
- To propose new signal processing techniques aimed at improving the robustness of hand-crafted and deep features against adverse conditions. Such techniques should be

useful for traditional approaches based on the training of classifiers with relatively few data and for modern ones based on deep learning.

- Finally, to develop new deep learning models aimed at dealing with additional problems arising in common scenarios, such as the subjectivity of strongly annotated data in human-labeled datasets.

## 1.3 Structure of the thesis

The organization of the chapters of this thesis is as follows.

This document is divided into seven chapters, with one additional section for bibliographic references. In this first chapter, we provide a global overview of the research project, indicating the main goals of this thesis and the motivations leading to the contributions contained in it.

**Chapter 2** covers the background work concerning this thesis. Related work on pattern recognition techniques applied to audio signals will be discussed, providing an overview of the main machine learning algorithms employed in the experiments carried out throughout this thesis. Special emphasis will be given to presenting the state-of-the-art regarding sound event classification tasks, explaining the open problems in the domain, the taxonomies used, and the most common performance metrics. The databases employed during the development of the thesis will be also here described.

In **Chapter 3**, an introduction to the typical pre-processing steps carried out before feeding audio data to a sound recognition system will be presented. Then, the most popular features used in the literature for sound classification are listed. A sensitivity analysis of those features is performed, studying as well different alternatives for fusing their information in adverse multi-microphone scenarios. In this context, the robustness of such hand-crafted features and others more sophisticated extracted from deep neural networks will be analyzed.

**Chapter 4** presents one of the main contributions of this thesis, related to a novel strategy aimed at converting variable-length audio sequences to fixed-length feature vectors that can be conveniently used to train traditional classification models. The use of this representation, not only solves the dimensionality problem arising from variable-length sequences, but also creates a transformed high-level representation of the sequence adapted to the temporal regions where the events concentrate their activity. This property, in turn, makes the system more robust to weakly segmented audio data.

Following the idea of the previous chapter, **Chapter 5** translates the rationale underlying the proposed non-linear feature transformation on the temporal axis to the domain of convolutional neural networks. In this context, a new distance-based adaptive pooling layer is presented and included within the architecture of a well-known end-to-end deep learning system, showing the advantages brought by such layer under mismatches between the training and test data.

**Chapter 6** addresses the acoustic event detection task using convolutional neural networks in the adverse scenario of weakly annotated data and overlapping audio events, using short-term energy envelopes as event activity ground-truths. This implies changing from a classification-based system to a regression-based one, where energy envelopes are estimated

by the network to indicate the presence or absence of a given audio event.

Finally, **Chapter 7** discusses the main conclusions of this thesis, outlining future research directions and the applicability of the results derived from this work.

# Chapter 2

# Sound Event Recognition

## 2.1 Brief overview

In the early 90s, Bregman studied how humans extract information from a complex acoustic mixture identifying the multiple (if so) sources active in a scene [9]. This study inspired the work in [160], originating the field of Computational Auditory Scene analysis (CASA), which aims to computationally imitate what a human listener does on source separation. Within this context we can introduce the term *sound event* as a specific label describing a sound produced by a source in an auditory scene, where the sound event detection/classification (SED/C) are sub-tasks within the more general CASA field.

Perceptual studies have attributed the name of soundscape (the equivalent of landscape in the audio domain) cognition. Usually, soundscapes are formed by complex sound signals, which may contain overlapping sounds (polyphony) belonging to the same or different classes. During this thesis, the term sound event will refer to any sound that can be categorized as environmental sound.

The task of assigning a label to the whole acoustic mixture is usually called Acoustic Scene Classification (ASC). If we are interested in identifying the individual sounds occurring within the stream, we can distinguish between two different approaches: Sound Event Classification (SEC) and Sound Event Detection (SED). The first one deals with the classification of isolated sounds segregated from the input mixture stream, while the second one is aimed at determining accurately the onset and offset times of an event of interest. The nomenclature of such tasks can vary slightly; recognition can be considered as a synonym for classification in many works, and in some others sound events are referred to as well as audio events or acoustic events. Sometimes, the differences between scene recognition and event detection tasks may be blurred. For example, humans automatically recognize all the events present within a scene before identifying the general environment.

To distinguish between detection and classification tasks, we follow in this thesis a simple rule. If the task requires providing the temporal timestamps of the recognized events, we are dealing with a detection problem. In contrast, if the task only requires to assign a semantic label, the problem becomes a classification one. Throughout this thesis we will use the term Sound Event Recognition (SER) to refer to any of the above tasks (detection or classification), where the context will make clear to which of those it is referring to. Figure 2.1 shows schematically how all the terms discussed above are related.

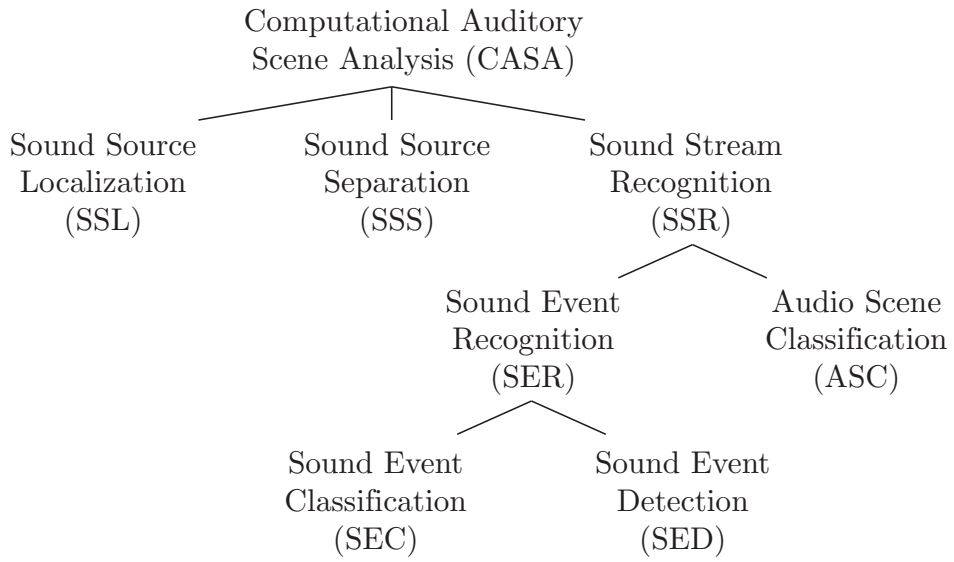



Figure 2.1: Taxonomy of the main tasks making up the CASA research field.

The popularity of the aforementioned tasks has increased over the years, thanks to a great extent to specific competitions such as the well-known *Detection and Classification of Acoustic Scenes and Events* (DCASE) challenges. In Figure 2.2, we can see the evolution of SED, SEC and ASC in terms number of published papers over the last 20 years, showing that the terms SED and ASC have gained considerable popularity in the last years. The first DCASE challenge was held in 2013 [138]. Figure 2.2 shows how from that year onwards the number of publications per year began to increase. The final results of the competition were presented in a special session in WASPAA2013 (*Workshop on Applications of Signal Processing to Audio and Acoustics*). The challenge was organized by the Centre for Digital Music, from the Queen Mary University of London, and IRCAM (*Institut de Recherche et Coordination Acoustique/Musique*) from Paris. At the very beginning, DCASE only comprised three tasks:

- Acoustic Scene Classification (ASC), with 11 participants.
- Sound Event Detection - Office Live (SED-OL), with 7 participants.
- Sound Event Detection - Office Synthetic (SED-OS), with only 3 participants.

These tasks are examples of challenges that a machine listening system should perform. The challenge was aimed at designing intelligent models able to recognize general sounds in everyday environments. For the scene classification task different indoor and outdoor locations in London were selected (street, restaurant, office, park, etc), while for the tasks of event detection only the office environment was selected. The difference between the live and the synthetic one was that the sound events were non-overlapped and overlapped, respectively. The number of overlapping events at the same time was one of the difficulties that the participants had to overcome. Another additional challenge, was the level of Signal to Noise Ratio (SNR) simulating the appearance of background noise.

From the submitted systems [138], the one giving the best performance for the ASC task was a support vector machine (SVM) classifier using MFCCs. While, for the SED task the winning system used Gabor filterbanks with a 2-layer Hidden Markov Model (HMM) classifier. The choice of using a statistical model like HMM for detection was motivated by their ability to model time series data. As per classification systems, SVMs seemed to be the most

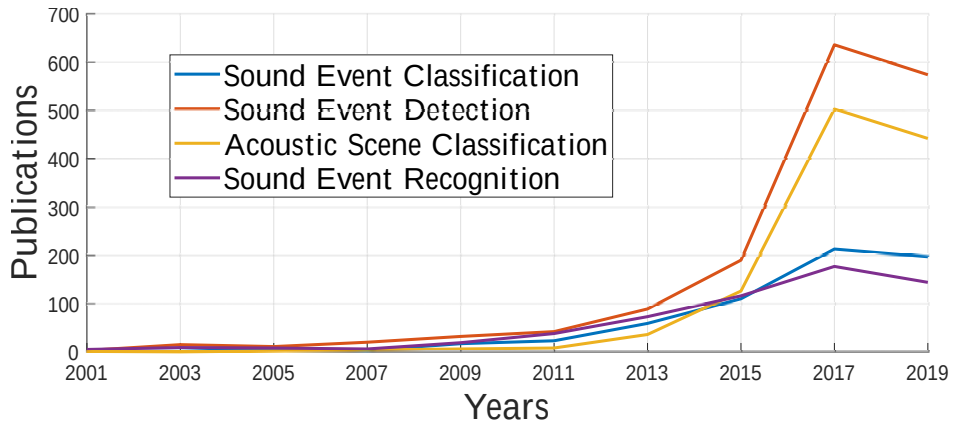



Figure 2.2: Number of publications per year on the topics of SED, SEC, ASC and SER, found in Google Scholar.

reliable one, being used by the top strongest systems in the challenge and outperforming the baseline system (accuracy of 55%), exceeding a mean accuracy of 70%. This motivated the use of the SVM classifier as a baseline model in the further development of this thesis.

The next DCASE challenge was 3 years later, maintaining the same tasks and adding a new one, referring to the topic of audio tagging. The challenges have been held in consecutive years to date. Given its popularity, the number of tasks as well as the number of collaborators and participants has grown over the years. Currently different universities and companies participate organizing this event, which demonstrates the interest shared in this topic both by academic and industrial entities.

## 2.2 Pattern recognition on audio

The process of training a machine to perform any given task, comprises a set of common steps. The standard stages comprise: to have a training set of data, to define a optimization criterion and to evaluate the system under a certain metric. The different machine learning techniques can be divided based on how the training data is fed to the model. What usually makes the training set different from the testing set, is that the data for training is formed by known samples which have been individually inspected and sometimes manually labeled. However, having labeled data is not always possible. Depending on the kind of training data available different approaches are defined:

- **Supervised**. The input data is made up of training samples with their corresponding labels. The model tries to approximate a mapping function which is valid for the unseen testing data [17]. Types of supervised problems can be grouped into classification (when we have categories, predicts to which category the data belongs to) and regression (when the output we desire to predict is a real value).

- **Unsupervised**. The system is fed with training data that does not have any target label [38]. Examples of problems requiring unsupervised methods are clustering (where it is wanted to identify the different groups in the hierarchical structure of the data) or association (to find any affinity relationships between the data).

- **Semi-Supervised**. A trade-off between the previous two. The system is fed with training data from which we have partial labeled outputs [38]. Real-life databases

usually have this problem, since it is expensive and time-consuming to manually label all the available data.

The work developed in this thesis is based on a supervised learning model, aimed at recognizing sound events, assigning one label to an audio signal. Sound events can be categorized into multiple categories, and the model we wish to train has to be aware of the range of classes we are interested on. In a closed-set scenario, where the number of classes to recognize is limited and is previously known, the classification problems can be divided into the following subsets:

1. Binary classification: this scenario is the simplest one, where only two classes are present and we have to decide whether it belongs to one or to the other. Speech/music discrimination is an example of this type of task [25, 40].

2. Multi-class classification: in this situation there are more than two categories involved. The restriction is that the given example has to belong only to one of the available classes, which means that they have to be mutually exclusive. This approach has been used for sound event classification, in works such as [142, 114, 95].

3. Multi-label classification: (also known as audio tagging), can be considered the most complex scenario. In this situation there are multiple classes but one sample can belong to more than one class at the same time [24, 98]. Audio tagging is becoming more popular since most of the every-day audio is polyphonic. Thus, one audio excerpt may contain sounds belonging to more than one class. One relevant example is music genre classification, where a given song could be Pop, and at the same time have elements from Electronic and Rock [106, 129].

On the other hand, the output of the learning model can be used in SED tasks. The aim of these applications is to identify the start and end time of the different sound events present in continuous audio streams. The problem is similar to the classification one, since we want to identify the sound of interest from a given audio clip. The difference is that, in this scenario the given audio excerpt is longer than before (varying from several seconds to tens of seconds), and the timestamps of the sound of interest have to be detected. In the early stages of SED, the solution was to perform *detection-by-classification*. Here, the sound excerpts were analyzed using a sliding window, classifying them first as silent/non-silent events (segmentation step) and secondly as belonging to a certain class (classification step) [142]. The main drawback of this approach is the selection of the window length, which will determine the final performance of the classifier, due to the highly variable duration of the sound events even within the same class. One solution is to analyze the acoustic stream frame-by-frame (the sliding window is set to a compromise frame-level length). The estimated probability outputs of the model are thresholded using a fixed value and the probabilities above that threshold are mapped to its corresponding label. Another technique is called *detection-and-classification*, where the detection step is applied before performing classification [112, 150]. Recent works using this technique rely on a time-frequency segmentation mapping [69] for separating background noise from the sound events. When background sounds are present while trying to identify an event of interest, designing a good detection algorithm becomes tedious, and the performance of the system will rely on how well the algorithm is able to detect the sound events. In subsequent chapters we will discuss the importance of a good detection technique and what methods can be applied to reduce its impact on the performance of the classification model.

The problems machines have to face when analyzing and processing acoustic signals in order to perform any classification task vary depending on the domain. Signals have to be

processed differently if the aim is to identify a given speaker or to recognize the instruments present in a given music track. Three of the most important domains in which we can divide machine listening tasks are the following ones.

### Speech

Sound communication has played a crucial role in the survival of species (from love calls to warning sounds against predators). The sense of hearing is well developed in mammal species, being capable to discriminate with a resolution of $10\mu s$ which ear is the first to receive an acoustic stimuli [21]. Humans have developed a unique way of communication with each other, known as speech. The development of spoken language has contributed to communicate complex information across generations. Speech processing comprises two main stages; generation of an acoustic speech signal (production) and recognition and understanding of the signal, what is known as perception. The multiple applications that automatic speech processing can offer have favoured this field to become the most popular domain. Spoken language has been studied by many researchers and involves a wide range of applications, such as automatic speech recognition (ASR) [158], speaker recognition [136] (identifying the person speaking) or speaker diarization (identifying the speaker among several people talking) [6]. In Figure 2.3 we can see how ASR systems have evolved from the 18th century up to current state-of-the-art systems.

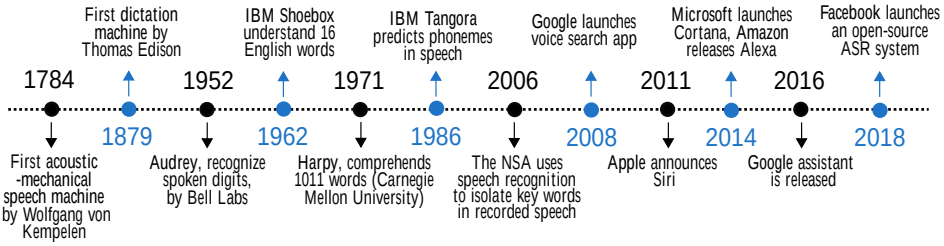



Figure 2.3: History of speech recognition technology

Many applications have been implemented in recent years, and thanks to deep learning huge progress has been made in fields like text-to-speech synthesis with the implementation of WaveNet [105]. Currently, the popular Google assistant service is powered by WaveNet. However, speaker identification systems provide still poor performance compared to other biometric techniques based on fingerprints or iris. There are many challenges to overcome with a wide open areas for research, such as detection of gender [85], age [18, 162], or emotions [140]. Possible applications are related to the creation of devices capable of identifying sleepiness or cognitive disorders [14] by only analyzing our speech.

### Music

Almost everybody enjoys listening to music. The digital revolution in music has helped to easily store and distribute music through the Internet. Most users consume tons of music everyday and others are even capable of creating it. Despite the popularity of the domain, the research field is relatively young, starting in the 1920s with the theremin instrument, where the player controls the oscillator with the capacitance of his/her hands. During the 1940s and 1950s composers created music using equipment from electronic labs thanks to the development of signal processing. Later, in the 1970s, computers began to be used as synthesizers to create new styles of music [122]. Within the music processing domain, researchers have addressed various aspects related to the complexity and diversity of music, such as the rhythm, genre, tempo, timbre and instrumentation. There are many applications,

including automatic music transcription (AMT) [15], instrument recognition [88], separation of the main melody from the accompaniment [45], automatic music tagging [146] or the most popular one, music information retrieval (MIR). Latest advances in the field of MIR (extraction of meaningful features from music), using deep learning [35], have helped to develop user music consuming applications, such as music recommendation systems or music browsing interfaces [131]. The first International symposium on music information retrieval (ISMIR) was held in 2000, bringing together every year researchers interested on processing, searching and accessing music-related data.

### Environmental sounds

A basic straightforward way of defining this domain is by relating it to those acoustic areas not belonging to speech or music signal processing. However, this domain has a more complex definition. A wide range of acoustic signals that can be categorized into multiple classes can be found within this domain. The audio research community became interested in the field with the evaluations carried out in 2006 as part of the CHIL project, where two databases comprising real environmental sounds were used in the tasks. Being the youngest domain, the amount of data released was scarce until few years ago, making it difficult to tackle properly the challenges present when analyzing environmental sounds. Some of the problems arising in this field are directly related to some of the tasks already mentioned, such as ASC [48], audio tagging [70] or SED, including special scenarios such as the consideration of rare acoustic events [41]. Figure 2.4 shows an example of ASC where the acoustic scene has been classified as office environment, from the same example the different sound events composing the signal are labeled in a second step. In A) the onset and offset times are marked for each example, while in B) the whole audio excerpt is matched to one or more labels by performing audio tagging. Systems implementing the aforementioned tasks can be used in many applications such as acoustic monitoring [159, 139], surveillance [153] and health care systems [151].

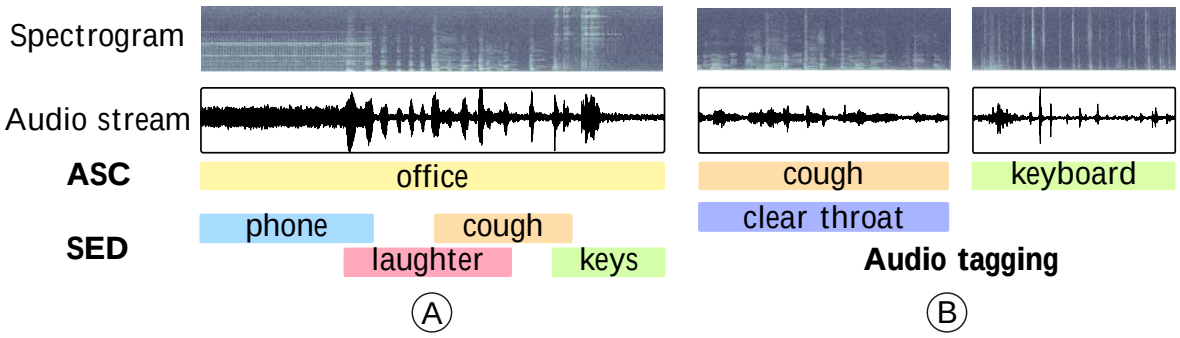



Figure 2.4: Representation of ASC, SED and audio tagging tasks.

## 2.3 Audio processing and recognition pipeline

This section provides an overview of common processing steps carried out by many SER systems, including the need to perform a temporal analysis of the input audio signals or the extraction of relevant discrimination features. Similarly, the common pipeline followed for creating SER systems will be dicussed.

### Audio representations

Sound is a vibration that propagates as waves through a medium, such as air or water. Sounds can be recorded using a electroacustic transducers like microphones. Until late 19th century, acoustic signals were recorded as mono sounds (coming from one position). Currently, most acoustic signals are recorded using two microphones as a stereophonic sounds, simulating the natural hearing. In Figure 2.5, a stereo waveform of a truck passing by is depicted. Differences between the left (1) and right (2) channels can be clearly seen, where typically the mean of both channels (3) is used for further analysis of the audio. Some works [60, 39, 2], use both channels as an input to the classifier in order to avoid the loss of information. A step further is the use of multi-channel techniques employing signals impinging on microphone arrays, which allows to capture more faithfully spatial information. The multi-channel signals can be processed in different ways, aggregating the resulting information by means of a variety of fusion techniques with the aim of characterizing better the acoustic signal [94].

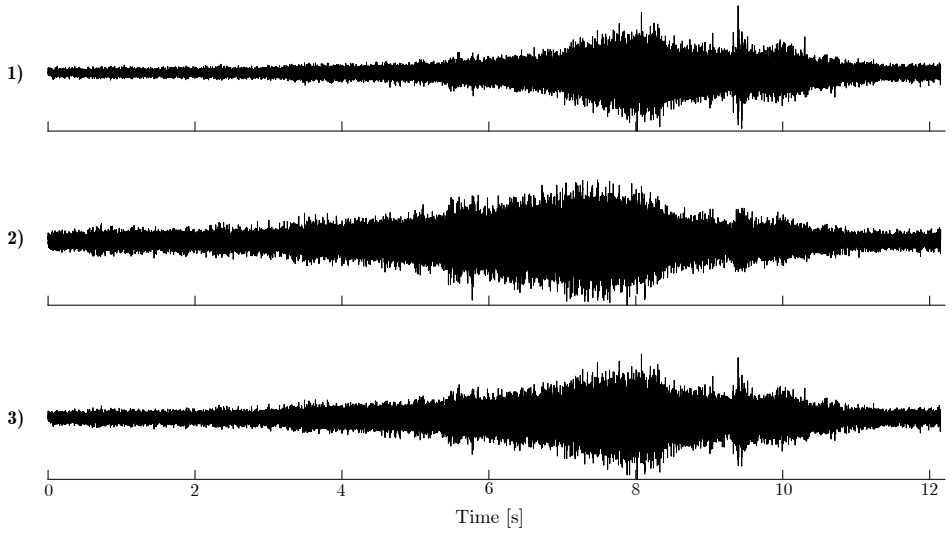



Figure 2.5: Waveform of a truck passing. 1) Left channel. 2) Right channel. 3) Average of both channels.

Trying to classify sounds using only their waveform with traditional learning methods is nearly impossible. The relevant properties of the signals are not sufficiently enhanced in their raw waveform and the dimensions of the input are not appropriate for traditional classifiers even if a low sampling frequency is used. Thus, several representations have been proposed in the literature in order to characterize sound events. Most of these representations try to mimic human perception properties. For example, the use of the mel scale accommodates the non-linear frequency resolution of the human hearing system.

A well-known general time-frequency representation of audio is the spectrogram, which is obtained from its short time Fourier transform (STFT). Figure 2.6, shows two different waveforms and their corresponding spectrograms, each of them with different structure, where the applause has a very stationary behaviour in contrast to the periodicity of the envelope of the phone ringing tone.

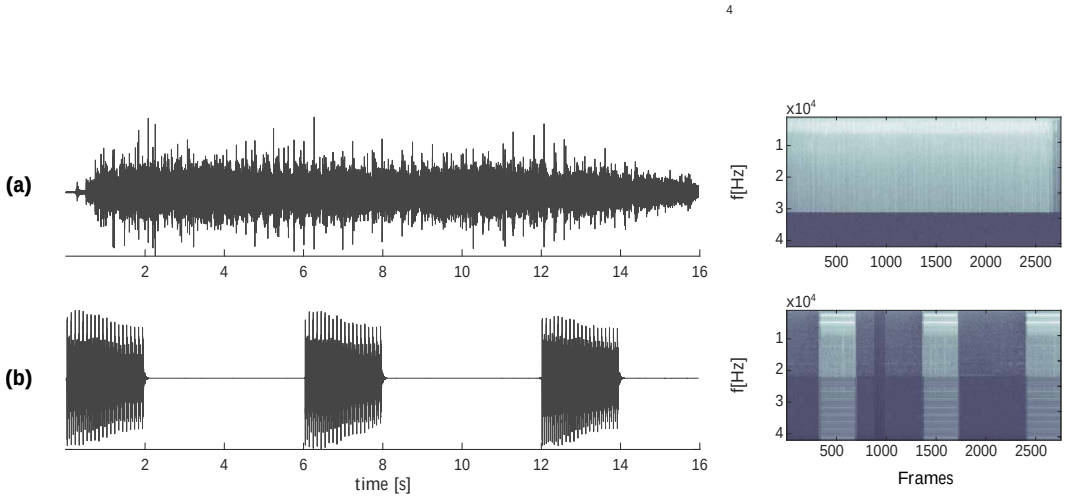



Figure 2.6: Representation of (a) an applause and (c) a phone ringing sounds.

### Temporal analysis

The processing steps must enhance properly the characteristics of the audio signal in order to facilitate discrimination at later stages. At the same time, they have to capture the temporal evolution of the sound. In fact, something that makes audio processing very challenging is precisely its temporal variability. Unlike images, audio signals change during time, which makes its main parameters time-dependent.

Some learning models do not allow a variable input, like for example SVMs or feedforward neural networks. Therefore, a fixed length conversion has to be conducted when using these models with sequences of variable length. Although the models could be used considering a frame-by-frame approach, where each input sample would correspond to one frame of the total audio input, this would also increase significantly the size of the total number of samples per class. Moreover, the problem with this approach is that the sequential information of those frames would be lost, resulting in poor performance. This is primarily due to the fact that one single frame is not enough to represent properly the characteristics of an audio category.

General approaches to analyze the audio signal can be divided into *short-term windowing* and *mid-term windowing*. In order to get a better characterization of the signal, short-time analysis considers the signal to be stationary during a short interval. A window function is used to emphasise the signal values during this interval, being the length of the window a trade-off parameter. It has to be sufficiently long to capture the values of interest but it needs to be short enough to ensure a reliable measurement, since the signal is considered to be quasi-stationary within such interval. Typical values for speech sounds are window lengths of 20-40 ms. For music analysis, the values increase up to 50-80 ms, while for environmental sounds the window length has to be selected as an intermediate value given the highly-variable length of the events. Some sounds are shorter than others, such as door slam, gun shot or glass break, compared to the longer ones like baby crying, music or sirens. Therefore, it is important that these changes are reflected in the feature extraction step. The second approach is mid-term analysis, here the audio signal is first divided into mid-term segments (from 1 to 10 seconds, depending on the domain), then the short-term analysis is carried out for each segment. From the short-term sequence computed within each mid-term segment, a set of feature statistics is extracted, assuming homogeneous behaviour during this interval. For example, for music genre classification, it has been observed that extracting

short-term features and later averaging them using mid-term windowing (to extract accurate statistics of the signal) has given better performance results [147].

Inspired by the mid-term analysis and the *attack-decay-sustain-release* (ADSR) model for music signals, a number of fixed temporal sections can be selected after the short-term analysis. To overcome the information loss due to the fix-length conversion, statistical parameters (such as mean and standard deviation) are extracted from three mid-term sections, named onset-midset-offset. For each of the short-time features, the mean and standard deviation across each of the three uniform sections are computed. In Figure 2.7, the short-term spectrogram (b) is computed from a waveform (a) of a *truck passing by* class. Then, three uniform sections are defined, marked with red dashed line. The mean for each frequency bin across the fixed mid-term sections is obtained in (c). This approach will be used during some of the experiments of this thesis when training a SVM, setting a fixed length input for all the audio examples from different databases. To mitigate the loss of temporal information, the dynamics (delta coefficients) of the signal will be stacked with the rest of features.

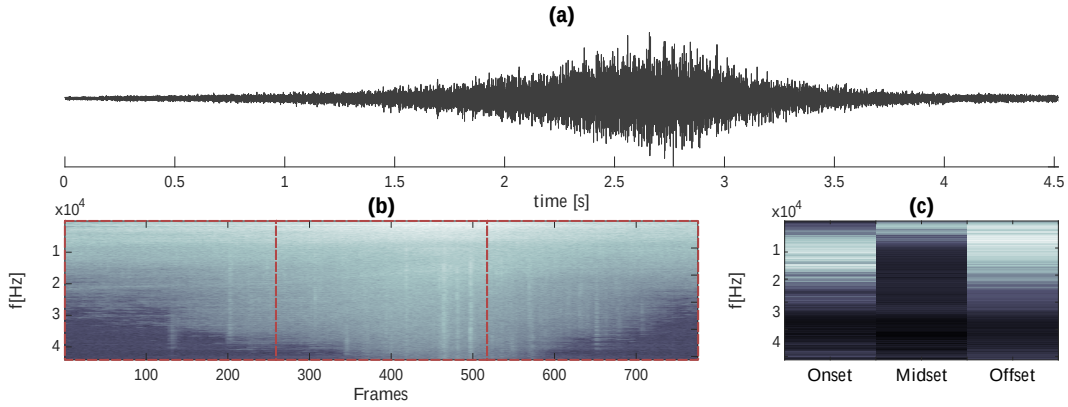



Figure 2.7: Waveform with its corresponding short-term and mid-term analysis.

### Feature extraction

Audio feature extraction can be seen as a data rate compression, where only relevant information of the signal must be kept, discarding non-informative data. How to extract such information relies in a series of audio processing parameters. Traditionally, features were originally designed for speech recognition or music retrieval. This is the case, for example, of log-mel spectrograms and MFCCs, which have also been applied to SER tasks. However, environmental sounds are mostly non-stationary, and many of them do not show meaningful patterns or structures such as speech. Therefore, the use of a single set of features may not be useful to capture relevant information from many types of events.

The spectrogram is usually too generic and complex patterns identifying the events may not be substantially enhanced. Commonly, a series of features have been used together in order to obtain a better representation of the variable nature of sound events [42]. This step is usually known as feature engineering. However, as discussed at the beginning of this chapter, selecting an appropriate set of features depends on expertise and the final application. The output is usually a wide combination of hand-crafted features storing discriminative information from the training examples. However, they sometimes do not generalize properly because they tend to be too specific [138]. Below, we list some of the most popular hand-crafted features:

- MPEG-7 toolkit [26] was an advancement for sound event related task, because it helps to introduce a common framework for sound event descriptors. The interface includes low-level features, such as sharpness, signal power, slope, spectral flatness and fundamental frequencies [161]. Another feature based on Audio Spectrum Bases (ASB) added in the MPEG-7 is called Audio Spectrum Projections (ASP) [65].

- Mel-Frequency Cepstral Coefficients (MFCCs) [30] represent the Discrete Cosine Transform (DCT) of the log-spectral power of the mapped signal using the nonlinear mel frequency scale. The zero coefficient represents the mean energy while higher order coefficients are usually discarded, obtaining a compact representation of the envelope. They had been broadly used for ASR [156] and also successfully incorporated in music related tasks from 2000 [91]. Their application in different audio domains is still popular given their great capacity to capture global spectral properties.

- Mel-Frequency Energies (MFEs) also known as filter-bank energies (FBE) [108], have gained popularity over MFCCs thanks to deep learning techniques, since deep learning systems tend to perform better when having as input the log mel spectrogram (without the need of the decorrelation step added by MFCCs).

#### Feature learning

The increase in the number of acoustic examples available in recent databases have favoured the development of feature learning techniques. These representation techniques are aimed at solving the problematic of choosing the proper set of features for sound event recognition among the wide variety of possible hand-crafted features. Some works have followed this approach for solving ASC and SED tasks, such as Salamon et al. [125], where clustering techniques were used to learn dictionaries or codebooks from representative examples, or the ones using the bag-of-features approach [117, 166] to learn the intrinsic representation of the sound events.

In general, we have mentioned that audio raw representations are not suitable for feeding directly the classification model, even with a fixed-length, because of its considerable dimensionality. A part of having a extremely large dimension, the waveform cannot represent perceptually similar sounds as being close neighbors in their vector space. However, recent advances in deep learning techniques [16] particularly in CNN, have motivated some researchers to directly use the raw audio signals as input to the neural network. This has been tested mostly in ASR problems, like in [107], where their CNN model trained with raw data outperformed the ANN model with standard cepstral features as input. Similar, in [60], a CNN model using multichannel waveforms obtains a good feature representation by trying to imitate an auditory filter-like representation.

#### Recognition pipeline

An overview of the classification learning pipeline can be observed in Figure 2.8. During the supervised learning phase, the features are extracted from the training data and are given to the classifier as input combined with the ground-truth labels. Once the acoustic model is built, new testing data will be given to the model in order to estimate its label during the test phase. This conventional scheme is the one followed during the development of the thesis.

In general, the recognition of audio events can be performed either sequentially or by following a segmentation-based approach. Sequential or online methods analyze the continuous audio stream using sliding windows that may range from tens of milliseconds to few

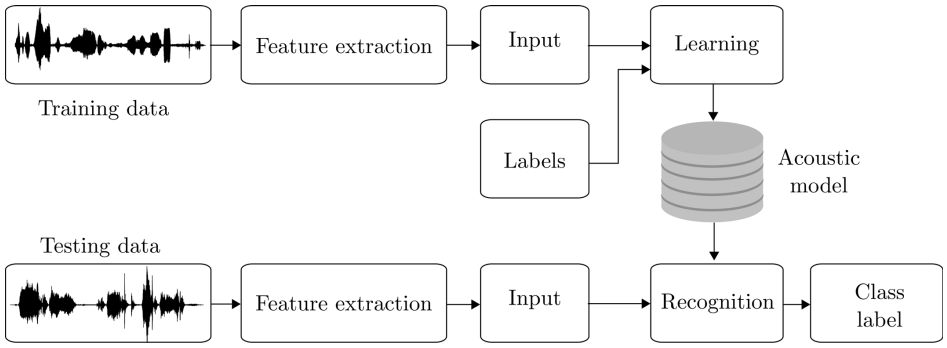



Figure 2.8: Sound Event Classification learning pipeline.

seconds, depending on whether they use context information or not. On the other hand, segmentation-based or offline approaches rely on a pre-processing step that estimates the onset and offset times of potential events, feeding the classifier with a sound excerpt that contains the whole event duration. Figure 2.9 shows how the information extracted from the sliding windows of the sequential approach (a) is aggregated by the model to predict the corresponding label. For the segment-based (b) approach the sound events have been previously isolated and provided to the model, which will predict one label per segment.

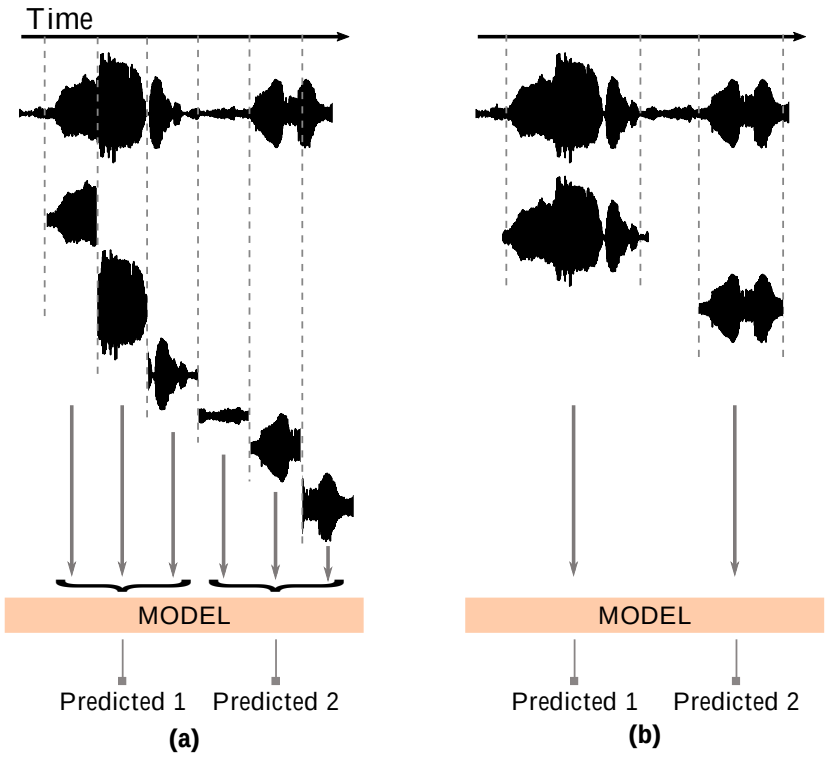



Figure 2.9: (a) Sequential (online) and (b) segmentation-based (offline) approaches.

Sequential methods using sliding windows encompass well-known processing schemes such as those based on Bag-of-Frames (BoF) [11], where feature statistics from a set of temporal frames are usually modeled by a GMM for each class. At test time, the GMM estimates are summed over all frames, choosing the class with highest likelihood. The sliding window approach has also been applied in recent deep learning (DL) approaches using CNN [12], where features are automatically learned from raw audio data. The final class labels are usually

obtained by averaging the predictions across all windows.

In segmentation-based methods, the classifier is assumed to be fed with the whole event sequence. Common approaches of this type are those based on HMMs, where the temporal evolution of the events is inherently taken into account within the model. General schemes use a fixed number of states for each event, detecting a given event by exploring the Viterbi path passing through the different states. The outputs from each state are again usually modeled by GMMs of common low-level features extracted from the different frames. Alternative methods divide the input sequence into a set of overlapping windows, where each window is assumed to be related to a meaningful section of the event [120]. Then, statistics over each window are extracted to create the final feature vector used for classification [111, 50].

## 2.4 Evaluation metrics

The performance of the learning model needs to be measured using well-defined evaluation metrics. The first step is to split the available data into training (large quantities generally lead to better performing systems), testing (large amounts give reliable and more precise estimates) and if available an evaluation set (used once the system is tuned to evaluate the system performance on new data). The performance metrics are calculated during the training and testing phases in order to tune the parameters of the model. To avoid overfitting by using a particular and fixed training data and to help with the generalization properties of the model, the available data is usually divided into cross-validation folds. The performance is then iteratively calculated for each fold and the final system performance is obtained by averaging the results on each split.

There is no consensus about which metric is better to evaluate the overall performance of a classification/detection system for sound events. Metrics like accuracy, F-score, acoustic event error rate (AEER), receiving operating characteristic (ROC) or area under the curve (AUC) have been broadly adopted by the audio community as an standard for comparison purposes and evaluation rankings. In this section, we present some of these metrics that have been broadly adopted by the audio community.

Given a ground-truth class label corresponding to a test sample in the two-class case, we identify such sample according to the prediction of the system as follows:

- True positive (TP): The system correctly predicted that the sample class corresponds to the reference class.
- True negative (TN): The system correctly predicted that the sample class does not belong to the reference class.
- False positive (FP): The system erroneously predicted that the sample corresponds to the reference class.
- False negative (FN): The system erroneously predicted that the sample does not correspond to the reference class.

The percentage or rate of samples labeled according to the above definitions provides a set of statistics from which further performance measures can be obtained. The most-used metric is Accuracy ($ACC$), which measures the quotient between the number of correct

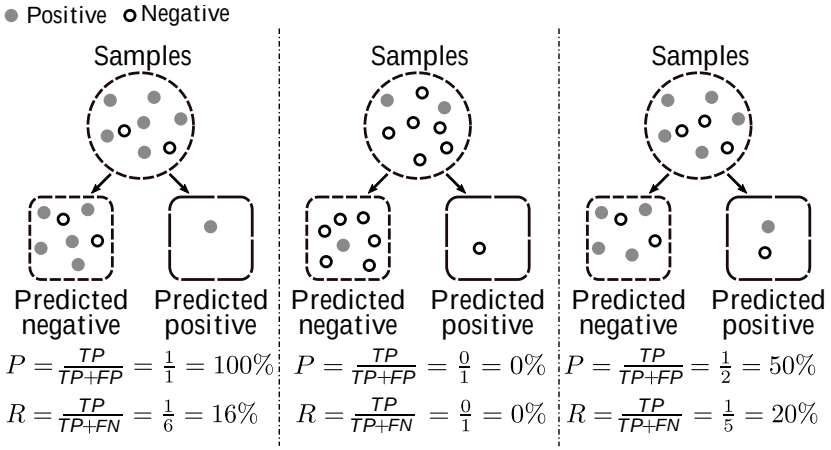



Figure 2.10: Three different classification problems showing the precision and recall metrics.

system outputs and the total number of outputs:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}, \tag{2.1}$$

where $TP$, $TN$, $FP$, $FN$, refer to the corresponding computed rates. The drawback of this measure is that if the data suffers from class imbalance, the TNs may dominate the accuracy value, making it not possible to rely on this measure. In the field of information retrieval, the term recall ($R$, sometimes referred as to Sensitivity) is widely used, which was defined in 1955 by Kent et al. [110]. Similarly, the term precision ($P$), as the positive prediction value was conceived. Both metrics are defined as:

$$P = \frac{TP}{TP + FP} \quad R = \frac{TP}{TP + FN}. \tag{2.2}$$

Having a precision of 1 does not ensure the good behaviour of the system, as can be seen in Figure 2.10. This only ensures that there are no false positives, but may be a high number of false negatives. For this reason, both metrics are usually combined together using the F-Measure (also known as F-score or $F1$). The measure was introduced in 1979 by van Rijsbergen [121] and is obtained as the harmonic mean of precision and recall by:

$$F = \frac{2TP}{2TP + FP + FN}. \tag{2.3}$$

When the reference label indicates class $a$ but the system outputs class $b$ instead, then we can consider this situation as a substitution. Some metrics, such as the ones used in the CLEAR evaluations [142], use this approach, where the AEER is defined as:

$$AEER = \frac{D + I + S}{N}. \tag{2.4}$$

The metric takes into account not only the number of events to detect ($N$), but the number of deletions ($D$, missing events), number of insertions ($I$, extra events) and the number of event substitutions ($S = min\{D, I\}$). The drawback of this metric is that it only considers a monophonic output (one class per sample), without taking into account the possibility of having multiple classes at the same time. For polyphonic mixtures, new metrics have been defined, where the measurement is done in fixed-length intervals or at event-instance level. Therefore, in a multi-class scenario, the aforementioned metrics can be calculated using the following approaches [100]:

- Frame-based or segment-based: were the size of the segment varies depending on the resolution needed in the final application.

- Event-based or instance-based: which usually adds a tolerance parameter to deal with the possible onset/offset ambiguities due to manual annotators.

When calculating the overall performance, the averaging options to aggregate the measured statistics are called *micro-averaging* (for the instance-based) or *macro-averaging* (for the event-based approach). In micro-averaging, equal weight is given to each individual decision, therefore the large classes will dominate over the small ones. On the other hand, when macro-averaging is applied, equal weight is given to all the classes, being mandatory the presence of all the classes.

## 2.5 Sound taxonomy

Soundscape research became popular in the 1990s but the term was first introduced by Schafer in the 1970s [130]. During the years, different definitions have been used to describe the concept of soundscape, currently it is defined as "the acoustic environment as perceived or experienced and/or understood by a person or people, in context" by an ISO working group [62]. By definition, the dependence on the context when categorizing sounds is clearly stated. When identifying sounds within a given scenario it is important to have an organized categorization of the sounds. If we want machines to understand the soundscape the same way humans do, first we need to establish a categorization framework. To this end, taxonomies (hierarchical structures of entities) and ontologies (modeling a wider range of relationships between entities), have to be defined.

One of the first categorization for classification of environmental sounds was defined in 2000 by Marcell et al. [92]. A group of college students were selected to freely describe a collection of 120 environmental sound. After an agreement process, 27 categories were obtained. Such categories described the sound source (animal, human, insect..), location (bathroom, nature, household, ...) or actions (accident, sickness, sleep,...) among other properties. Different categorizations have been defined, trying to establish a common ground-truth. For example, Brown et al.[20] divide the categories in a hierarchical structure; on top there are defined the places (urban, rural, wilderness and underwater) then for each place there are different sound sources (humans, nature, animals, transport, ...). Finally each source has also a more detailed one. For example, in nature we have wildlife, wind, water thunder and earth movement.

Some approaches have tried to define more specific context taxonomies, such as the one by Salamon et al. [128] categorizing sounds happening exclusively in an urban soundscape. An example of the higher levels can be seen in Figure 2.11. The lower levels contain categories corresponding to noise complaints of New York City. Finally, the most recent and extensive hierarchy for sound events is given by the AudioSet ontology [46], covering a total of 632 classes.

However, these taxonomies also suffer from mismatches among them. Above we show a comparison of the two aforementioned taxonomies for environmental sounds. For example, the sound event *Police*, inside the Urban Sound taxonomy, has the following hierarchy

- Mechanical > Motorized Transport > Road > motorcycle > Police

whereas for the Audioset ontology (which takes into account more complex relationships) has two different meanings:

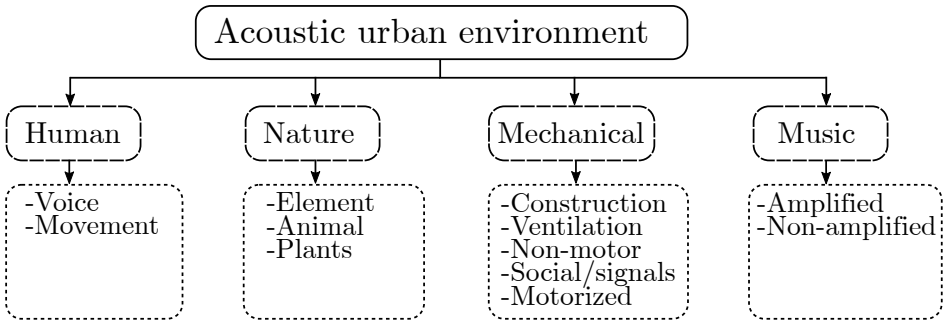



Figure 2.11: Subset of urban taxonomy from *Urban Sound* [128].

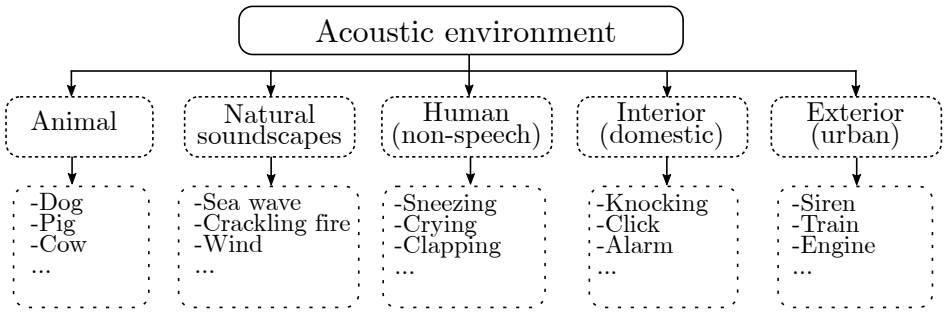



Figure 2.12: Subset of environment sound taxonomy extracted from the *Environmental Sound Classification* (ESC) dataset [115].

- Sounds of things > Vehicle > Motor vehicle (Road) > Emergency vehicle > Police car (siren)
- Sounds of things > Alarm > Siren > Police car (siren)

Another example of sound event categorization can be found in [115], where a small part of their taxonomy is represented in Figure 2.12. Since the categorization of the events is quite dependent on the application domain, based on sound source(s), action(s) and context(s), an example of a possible sound event categorization given its context can be seen in Figure 2.13.

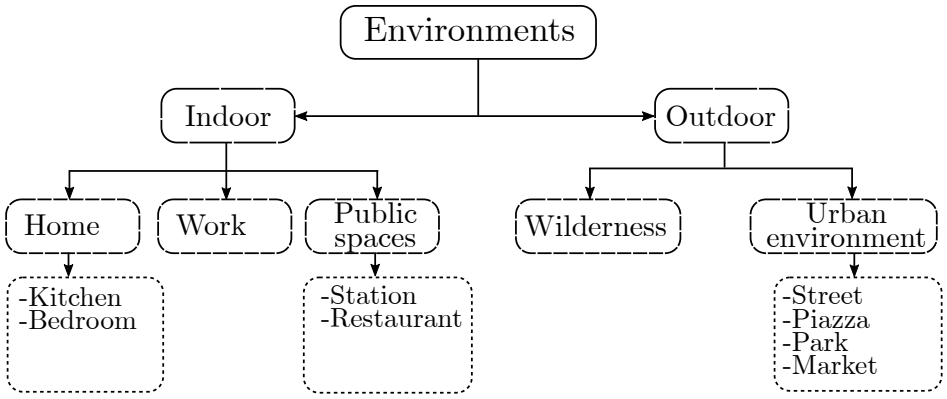



Figure 2.13: Example of taxonomy of contexts, extracted from [54].

The above discussion demonstrates how variable the interpretation of sound events can be. Having a common framework for soundscape taxonomy will help to communicate better across different domains. As it happens with the need for having reference performance measures to compare model performances with different databases and situations, it becomes

necessary as well to have a reliable taxonomy indicating that the categories evaluated by different models are actually the same.

## 2.6 Datasets

When evaluating the system performance it is also important to have a high-quality dataset. This means that the dataset should take into account the following properties: be extensive to include the maximum number of examples, cover all the possible categories and provide enough variability to ensure robust modeling. Particularly, for environmental sounds, covering all the possible sound events is quite challenging, since there are not well-defined categories, as has been previously discussed in Section 2.5. Besides, for an individual category, a sound may have different properties such as distinct source material or a varying production mechanism. For these reasons, building a database for this domain is a complicated task, being the annotation part one of the most expensive ones in dataset construction in terms of time and resources.

In this section, we describe all the datasets used during this thesis, having each of them different properties, mostly in terms of size and number of classes. The two first datasets are provided by the Queen Mary University of London, the third one is provided by Tampere University of Finland, the three of them being part of the DCASE challenges. The following two are given by the New York University and, as in the last one (ESC-50), their examples are extracted from Freesound. A small graphical summary of all datasets comparing their sizes is depicted in Figure 2.14.

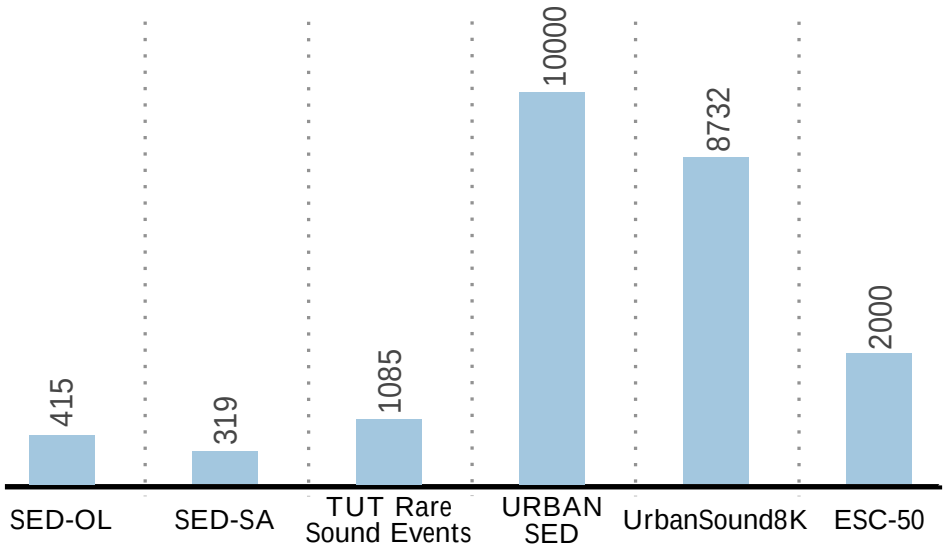



Figure 2.14: Total number of examples per dataset.

### Sound Event Detection - Office Live

The SED-OL dataset was created in 2012 and was used in the first DCASE challenge of 2013. The audio files are recordings of non-overlapped events from a real working office environment in the infrastructures of the Queen Mary University of London. The dataset is divided into 3 subsets: developing, training and testing. The training set contains stereo recordings of isolated events from 16 classes, which are: *alarm, cough, drawer, keys, laughter, page-turn, phone, speech, clear-throat, door slam, keyboard, knock, mouse, pen drop, printer and switch*,

with varying duration from 60 ms (switch) up to 30 seconds (printer). There are 20 examples per class, having a total of 320 examples for the training set. The development set consists of three scripted recordings almost 2 min long containing non-overlapped acoustic events in an office environment. All the examples are manually annotated, performing evaluations using an average of two annotators per sample. The dataset is publicly available online[1].

### Sound Event Detection in Synthetic Audio

SED-SA dataset, was released in 2016 and was used in task 2 of the DCASE challenge on the same year. The dataset is divided as the previous one into three sets, train, development and evaluation. The training set consists of 20 recordings of isolated (non-overlapping) sound events from 11 different office-related classes: *clearing throat, coughing, door knock, door slam, drawer, human laughter, keyboard, keys (put on table), page turning, phone ringing, and speech*. The total amount is 220 examples of monophonic files. The development set consists of synthetic mixtures with varying parameters, different event to background ratio (EBR) of $\{-6, 0, 6\}$ dB, different degree of overlapping and varying number of examples per class. The original unmodified dataset is publicly available online[2].

### Rare Sound Events

TUT Rare Sound Events 2017, was used in the task 2 of the DCASE challenge on the same year. The training set consists of 335 isolated events from three classes: *baby crying, glass breaking* and *gun shot*, having the sound events different duration. The development set consists of synthetically created mixtures long 30 seconds each, where each of them may or may not contain one of the three event categories. The mixtures do not contain overlapping events, but the background noise are recordings from 15 different audio scenes, which may contain artifacts and unwanted events. The audio scenes were extracted from the TUT Acoustic scenes 2016 dataset and the isolated events from Freesound. The resulting mixtures were modified to simulate different levels of EBR $\{-6, 0, 6\}$ dB. The complete dataset is publicly available online[3].

### UrbanSound8K

UrbanSound8K was released in 2014, contains 8732 audio examples all extracted from Freesound. The data downloaded from the online repository was manually checked by listening to it and saved only the ones containing the sound class of interest. The resulting audio files were manually annotated and limited to a duration of 4 seconds, for the short ones, zero-padding is added at the end. To select the sound classes, a study about noise complains in New York city resulted in a detailed taxonomy. For detailed definition please refer to [128]. Following the semantics of this taxonomy, 10 urban sounds were selected: *air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, and street music*. The dataset is pre-arranged in 10 folds for reproducibility purposes, from folds 1 to 6 are used for training, 7 and 8 for validation and the rest, 9 and 10 for testing. The dataset is publicly available online[4].

[1] http://c4dm.eecs.qmul.ac.uk/sceneseventschallenge/description.html

[2] http://www.cs.tut.fi/sgn/arg/taslp2017-crnn-sed/tut-sed-synthetic-2016

[3] https://zenodo.org/record/401395

[4] https://urbansounddataset.weebly.com/urbansound8k.html

### URBAN-SED

URBAN-SED dataset was released in 2017 and consists of 10000 synthetically created soundscapes. The software used for creating such dataset is called Scaper [127], an open source Python program. Scaper works as follows: given a certain specification the software creates mixtures and also returns their corresponding annotations. Some of the parameters are maximum number of overlapping events, SNR level, duration of the mixture, categories of the events and it is possible to specify the background noise by providing an audio file of the same duration than the mixture. The URBAN-SED dataset, following the previous one, is divided into 10 stratified folds and has the same number of classes belonging to categories related to typical urban noises obtained from the noise complains in New York city. The background noise used to create the mixtures is of Brownian kind, which tries to imitate the typical "hum" noise which is common in bus cities. The dataset is publicly available online[5].

### ESC-50

ESC-50, was released in 2015, and as the previous ones, all the audio files are extracted from the Freesound repository. The difference is that this dataset is focused on a higher variability of environmental sounds, not only in urban noises, resulting in a wider collection of categories with a total of 50 classes. These categories are divided in five main groups: *Animal sounds, natural soundscapes, human (non-speech) sounds, interior/domestic sounds and exterior/urban noises*, for a complete list of the 50 classes please refer to [114]. All the audio files are limited to 5 seconds long (padding zeros to the end if necessary), and the total amount of examples is 2000, equally balanced among the classes (40 examples per class). The dataset is publicly available online[6].

## 2.7 Classification models

This section briefly describes the learning models used throughout this thesis to perform recognition tasks. Classification models can be roughly divided into discriminative (models aimed at finding boundaries between classes) and generative (models aimed at characterizing the statistical distribution of the training data from each class). In this thesis, we will focus on the discriminative classification models. In particular, in the following two subsections we briefly outline support vector machines (SVMs) and deep neural networks (DNNs). Moreover, in the last subsection we also include a short account on the topic of transfer learning which is of key importance for the contributions in this work.

### Support Vector Machines

Support Vector Machines (SVMs), were introduced in a conference paper [19] for the first time in 1992 for a broad audience. But the concept of support vectors was born several decades before, when the field of *statistical learning theory* was introduced by Vapnik and Chervonenkis [154]. Several modifications and refinements were introduced in the following years. Worth mentioning are the introduction of the so-called soft margin [29], the reformulation of the problem that lead to the $\nu$-SVMs [133], and the generalization of the model to tackle regression tasks [155]. The first work showing the potential of SVMs on audio classification [55] was in 2003, solving a multi-class classification problem with the Muscle-fish

[5] http://urbansed.weebly.com/

[6] https://github.com/karoldvl/ESC-50

database [164], which consisted of audio files from sound effects and musical instrument sample libraries (from 1 to 15 seconds long).

The use of SVMs for classification comes mostly motivated by the scarcity of strongly annotated training data. Using the labeled data, the linear discriminative model introduces hyperplane separators. If the data is separable or quasi-separable (has some noise) the division is done in the original space of the input example, while if the examples are not linearly separable in the original space the division is created in a transformed higher-dimensional space (kernel feature space). As we shall see further on, the search for the hyperplane division in these transformed spaces, normally of very high dimension, is done implicitly using the so-called kernel functions, also widely referred to as the kernel trick.

When using SVMs for solving a classification task, we consider the following assumptions:

1. Transforming data into a high-dimensional space may convert complex classification problems into simpler ones that can be solved using linear discriminant functions.
2. Training patterns close to the decision boundary are the ones that provide the most useful information for classification.

Consider the simplest case scenario where two classes are linearly separable, as in Figure 2.15, where each feature vector $\mathbf{x}_i \in \mathbb{R}^n$ is represented as a point in an $n$-dimensional space ($n = 2$ in the figure). Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ be a function that separates the two classes, i.e. that outputs positive or negative values for each of the two classes. The corresponding separating hyperplane in $\mathbb{R}^n$, $H_0$, can be defined as:

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} - b = 0, \quad \mathbf{w} \in \mathbb{R}^n, \tag{2.5}$$

where $\mathbf{w}$ is a normal vector to the hyperplane, and $b$ is a bias parameter that determines the offset of the hyperplane from the origin. Without loss of generality, we can assume that the maximum and minimum value for negative and positive examples in the training set are -1 and 1, respectively. Therefore, we have two other hyperplanes, $H_{-1}$ and $H_1$, parallel to $H_0$ that are defined by $f(\mathbf{x}) = -1$ and $f(\mathbf{x}) = 1$, respectively. Under this setting, the distance separating these two hyperplanes, $\frac{2}{\|\mathbf{w}\|}$, is called the *margin*, and it can be considered as a good measure about how much $f$ separates classes in the training data. The word margin is also commonly used to refer to the whole region that lies between the two hyperplanes $H_1$ and $H_{-1}$.

The set containing all training vectors, $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L\}$, and the corresponding labels $Y = \{\ y_1,\ y_2, \ldots,\ y_L\}$, $y_i \in \{-1, +1\}$ constitute a particular instance of the learning problem. A given training set, $X$, with labels $Y$ is separated (in classes) by the hyperplane $H_0$, if and only if the elements of $X$ and $Y$ satisfy the following condition:

$$\begin{aligned} f(\mathbf{x}_i) &\geq +1, \text{ for } y_i = +1 \\ f(\mathbf{x}_i) &\leq -1, \text{ for } y_i = -1 \end{aligned} \tag{2.6}$$

That is, we can express that the data is separated as a set of restrictions. Moreover, and in order to obtain a good generalization, the margin must be maximal [19]. Therefore we can write the problem of finding the best separating hyperplane as an optimization problem with restrictions whose easiest formulation leads to a quadratic program with linear restrictions whose *primal* formulation is defined as:

$$\begin{aligned} &\text{minimize } \frac{1}{2}\|\mathbf{w}\|^2 \\ &\text{subject to } \quad y_i f(\mathbf{x}_i) \geq 1, \forall i. \end{aligned} \tag{2.7}$$

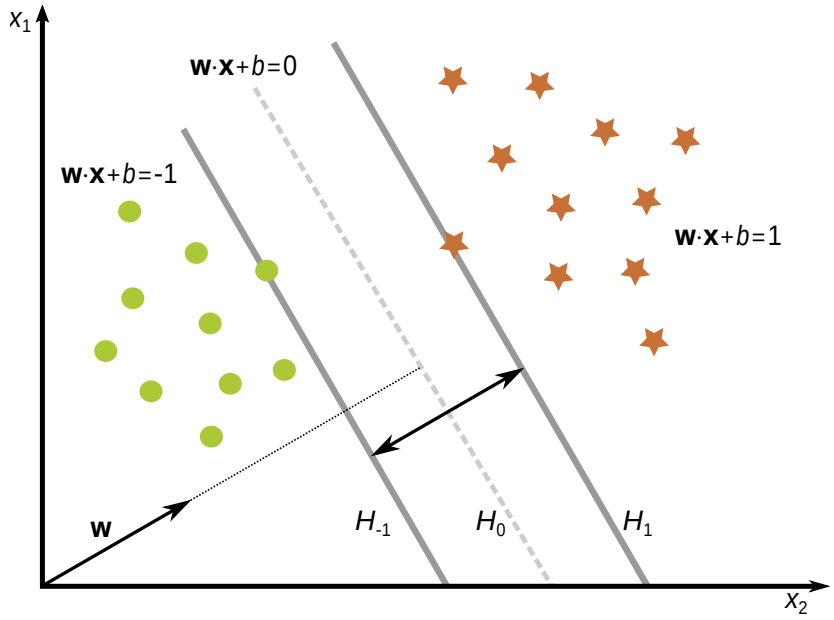



Figure 2.15: Representation of an optimal hyperplane with the weight vector and the bias.

From the above problem we can derive the corresponding Lagrangian which is

$$L(\mathbf{w}, b) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{L} \alpha_i (y_i f(\mathbf{x}_i) - 1), \tag{2.8}$$

and then by differentiation arrive at the corresponding dual problem

$$\begin{aligned} &\text{maximize } (\sum_{i=1}^{L} \alpha_i - \frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{L} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j) \\ &\text{subject to } \sum_{i=1}^{L} \alpha_i y_i = 0 \text{ and } \alpha_i \geq 0, \forall i. \end{aligned} \tag{2.9}$$

The dual problem is much easier to solve since in practice we will deal with the case $n \gg L$. Moreover, and as a consequence of the optimization of the Lagrangian and the Karush-Kuhn-Tucker (KKT) conditions we have the following facts once we obtain the optimal solution:

- The weight vector, $\mathbf{w}$, can be expressed as a linear combination of the training examples, where the coefficient corresponding to $\mathbf{x_i}$ is the dual variable, $\alpha_i$.
- Dual variables corresponding to training samples for which the absolute value of its output is above 1, must be zero.

This implies that the finally optimal hyperplane, which is given by $\mathbf{w}$, will depend only on training samples for which the output is either 1 or -1, e.g. the ones touching hyperplanes $H_1$ and $H^{-1}$. That is why these samples are called the support vectors.

The dual problem can be solved using standard quadratic programming techniques, being computationally affordable even for a high dimensional problems. Nevertheless, there exist a number of specific solvers for the SVM-related optimization problems [116].

When working with real data it is possible, due to noisy measurements, to encounter outliers, these are samples which are in the wrong side of the ideal separating hyperplane. In

other words, we may face in practice non linearly separable problems. In such cases, these outliers can be taken into account as misclassified data by softening the decision boundaries introducing a set of slack variables, $\xi_i$, and an overall penalty, $C$, which controls the trade-off between margin optimality and amount of misclassified examples. This directly affects the width of the resulting margin. As now we may have samples of the same class at both sides of the margin, we refer to this as the soft margin case. If $C$ increases, then the generalization capacity degrades, since fewer training errors are permitted, being $C = \infty$ equivalent to the hard margin case. The soft margin formulation of the optimization problem can be written as

$$\begin{aligned} &\text{maximize } (\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{L}\xi_i), \\ &\text{subject to } y_i f(\mathbf{x}_i) \geq 1 - \xi_i, \ \forall i \end{aligned} \tag{2.10}$$

After writing the corresponding dual formulation and considering the KKT conditions we arrive at very similar facts as before but now we can distinguish between two types of SVs, the ones laying inside the margin (for which $\alpha_i = C$) and the others lying either on $H_1$ or $H_{-1}$ (with $0 \leq \alpha_i \leq C$) which are called unbounded or free support vectors, being both involved in the definition of the optimal separating hyperplane.

A very nice aspect of the above formulations is that everything, including statements and solutions can be written only in terms of inner (or dot) products between pairs of vectors. This allows using the well-known kernel trick that consists of projecting the original data into a different space and then look for an optimal separating hyperplane in this new space. With this, SVM can be applied to problems that are not linearly separable but maybe so in other, more complex, and nonlinearly related feature spaces that can have very large or even infinite dimensionality. In particular the kernel trick consists of substituting all inner products in the new feature space by a convenient kernel function on the first space. This strategy can be applied in any Hilbert space with a corresponding inner product, $\langle \cdot, \cdot \rangle$.

More precisely, a transformation function $\Phi : X \rightarrow \mathcal{F}$ is defined in order to map each input vector, $\mathbf{x}$ with a point in the feature space $\mathcal{F}$. As an example, the criterion in the corresponding dual formulation will we rewritten as

$$\sum_{i=1}^{L}\alpha_i - \frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{L}\alpha_i\alpha_j y_i y_j \langle \Phi(\mathbf{x_i}), \Phi(\mathbf{x_j}) \rangle. \tag{2.11}$$

The dot product of the feature vectors in the transformed space, can be replaced by the kernel function:

$$k(\mathbf{x_i}, \mathbf{x_j}) = \Phi(\mathbf{x_i}) \cdot \Phi(\mathbf{x_j}), \tag{2.12}$$

that can be thought as a similarity measure between two data points in a new $\mathcal{F}$ space. The kernel functions have to fulfill the Mercer's conditions [101]. These conditions state that, every semi-positive definite symmetric function is a valid kernel, then there is no need to know the mapping function explicitly. Some examples of kernel functions widely used in SVMs are:

- **Linear**

$$k(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle. \tag{2.13}$$

Suitable when the number of features is large compared to the size of the data [32].

- **Polynomial**

$$k(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle^d. \tag{2.14}$$

  Popular kernel for natural language processing (NLP), having $d = 2$ as the most common degree, an special case of the quadratic kernel.

- **Radial basis function**

$$k(\mathbf{x}, \mathbf{x}') = \exp\left( - \frac{||\mathbf{x} - \mathbf{x}'||^2}{2\sigma^2} \right), \tag{2.15}$$

  where $\sigma > 0$.
  This kernel is the most widely used [55, 142] and its performance is generally comparable with the one obtained using polynomial kernel, however RBF is better for huge data training, due to its ability to adapt under many conditions.

- **Sigmoid**

$$k(\mathbf{x}, \mathbf{x}') = \tanh(\kappa \langle \mathbf{x}, \mathbf{x}' \rangle + \vartheta), \tag{2.16}$$

  where $\kappa > 0 \quad and \quad \vartheta < 0$.
  It introduces two parameters, $\kappa$ which scales the input data and $\vartheta$ that shifts the threshold of mapping.

If we want to successfully classify any given set of data, the choice of the kernel is a crucial part of the process, with requires a prior knowledge about the task. Other important parameter that affects the behaviour of the model is the selection of the parameter $C$ that controls the margin. The final computational cost of training a SVM model depends on the number of samples, as this number increases the number of support vectors that may grow linearly with it.

The SVM model is binary, being the design of multi-class models a further area of research. There are two popular and basic implementations to classify more than two classes. Given $n$ classes, $n$ two-class SVM models can be constructed to separate each class from the remaining ones, what is known as one-versus-all (OVA) approach. If instead, $n(n-1)/2$ two-class models are constructed for each of the pair-wise combinations, the approach is known as one-versus-one (OVO). For the former approach, the number of positive samples will likely be smaller than the negative ones. On the other hand, for the latter the computational cost becomes prohibitive if the number of classes is high. Other more elaborate multiclass extensions of the SVM model exist and have been used in the literature [36].

### Deep learning

Even though SVMs have been successfully applied in different audio-related tasks [55, 18, 145, 80], the use of the SVM as a classifier has its limitations. For example, their incapability to deal with non-static data, such as sequences or dynamic data or the lack of a direct formulation for multi-class classification. In the last years, approaches using variations of deep neural networks (DNN) for sound classification tasks have become extremely popular. One of the first works applying DNNs for environmental sounds was part of the EU funded project SMART[7]. The architecture used in [71] was a simple network with three hidden layers and was aimed at classifying four classes, performing slightly better than the SVM model.

[7] http://www.smartfp7.eu/

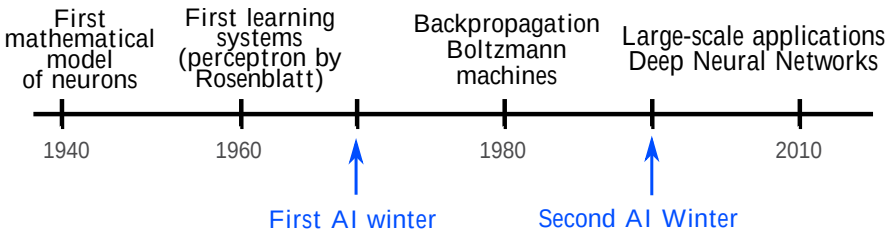



Figure 2.16: Evolution of Neural Networks within the Artificial Intelligence field through time.

Neural networks algorithms were inspired by the study of the most complex organ in the human body: the brain. The human brain is composed by billions of neurons, each neuron is connected using synapses to several thousands of other neurons. The concept of (artificial) neural network is almost 80 years old. In Figure 2.16, a scheme of the evolution of the neural networks subfield through time is shown.

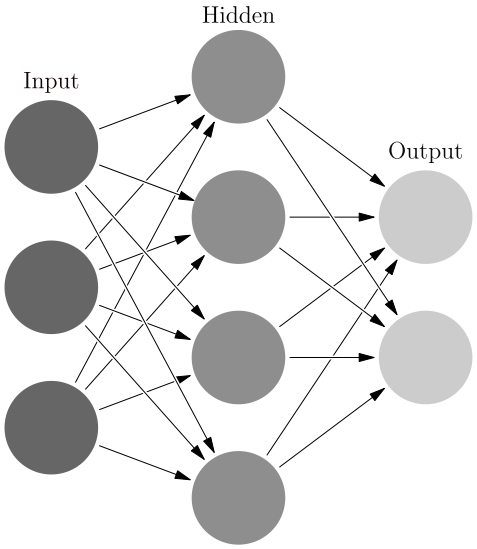



Figure 2.17: A minimalist example of a MLP architecture.

The first "deep" model was the multi-layer perceptron (MLP), defined as a sequence of layers (inter-connected artificial neurons), followed by a non-linear transfer function (also called, activation function). As depicted in Figure 2.17, an MLP is formed by at least three layers, a first input layer, one (or more) hidden layer(s) and one output layer. Each node is a neuron that has trainable parameters, that will be updated iteratively to minimize the error between the input and the desired output of the whole net. These neuron parameters are the weights of a linear function, including bias, that control the strength of the signal from a particular connection. The bias on the other hand, constitute a learnable offset that controls when and how a particular neuron passes its combined signal to neurons in the following layer.

The activation functions, applied at each of the neurons (except for the ones in the input layer), are used to allow for non-linearity and need to be differentiable functions in order to enable gradient-based optimization. Some of the most popular activation functions are the hyperbolic tangent, *tanh*, the rectified linear unit, *ReLU*, or the *Logistic* function, which are depicted in Figure 2.18.

The MLP presents some challenges that have to be taken into account when working with

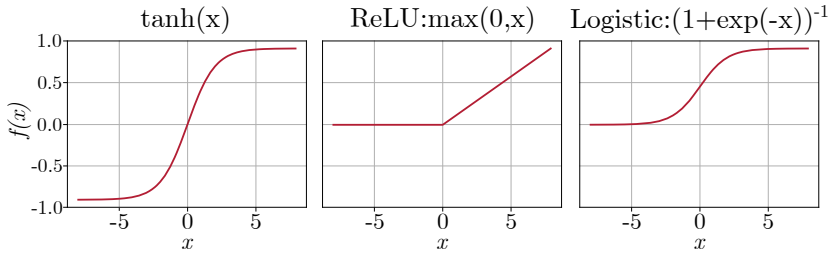



Figure 2.18: Popular transfer functions $f$.

audio data. The performance of the model depends of the input representation of the signal, similar to what happens to other classification models. However, this model is particularly sensitive to the scale of the data, something that has to be carefully addressed. The input data must have a fixed dimension, as it happens with SVM classifiers, which implies cropping or padding audio signals. Also, they are unable to exploit the temporal or frequency structure of audio data.

Further DNN architectures were studied in order to overcome the limitations of MLP models. Convolutional neural networks (CNN) were introduced in the late 80's for solving image classification tasks. They were successfully applied by LeCun et al. in [81] for character recognition, with an architecture of only three hidden layers. During the years, given their good results, these networks became very popular for improving image classification performance. The improvement in the computational capabilities of hardware allowed for deeper and more complex architectures such as Alexnet [72] in 2012, VGG [90] in 2015 and ResNet [56] one year later. Inspired by the results obtained in the image domain, in [58] Hershey et al. successfully applied a CNN classifier on a large-scale audio dataset, with excellent results.

### Optimization

Neural Network models are trained by means of an optimization algorithm. When a given input is fed to the first layer of the model the values are passed through all the layers until reaching the output layer. This step is called forward propagation. The difference between the resulting output and the desired one (also called target output) is measured using a certain loss function. These loss functions have to be differentiable and non-negative, since the algorithm used to update the network parameters is based on a gradient descent approach [123]. Depending on the model's goal, the loss function is selected to measure the discrepancy between the current behaviour of the machine and the desired behaviour on the training set. For example, if the problem to solve is a regression predictive task which involves predicting a real-value number, then an appropriate loss function would be the mean squared error (MSE). On the other hand, if the problem to solve is a classification task, then an adequate loss function is categorical cross-entropy.

Once the parameters of the network (weight and biases) are set, a gradient-based algorithm is used to find the minimum of the error function. This method uses a back-propagation procedure to calculate the gradient of each parameter by applying the chain rule from the output of the layer until the input layer.

### Convolutional networks

CNN architectures were inspired by the visual system's structure, thanks to the studies performed by Hubel and Wiesel in 1962 about the mammals' visual cortex[61]. In short, a

convolutional layer is exactly as an MLP layer but neurons are arranged in a particular 1D or 2D grid. Moreover, each neuron is connected to a particular local region (kernel) in the previous layer with the *same shared* weights accross the whole layer. In this way, and when appropriately set, the convolutional layer is equivalent to a convolution operation with a kernel over a given input. When the corresponding kernel parameters are appropriately learnt, this operation extracts local representations progressively more meaningful as we go deep in the network. Convolutional layers in a deep CNN are usually followed by subsampling or *pooling* layers that contribute to keep information redundancy low and contribute to the robustness of the finally obtained internal representations. By operating in this way, CNNs are able to detect an acoustic event regardless of its temporal position in the audio stream, which is very beneficial for SED task. The downsampled representation of the activations after a pooling layer is useful when the spectral position of a specific pattern belonging to a class may show small changes. The pooling operation after each convolutional layer in a CNN consists typically of a maximum or an average operation over a particular region in the previous layer with a given size and stride.

The activation function typically used is ReLU, which stands for Rectified linear unit (middle of figure 2.18), which is a non-linear function that outputs the input directly if positive, otherwise it outputs zero. This function is widely used since converges faster than other functions and avoids the so-called vanishing gradient problem. However, it has also drawbacks, such as the dead neuron, where biases and weights do not get updated because of very small values in previous layers. Alternatives to avoid this problem are the use of Leaky ReLU (instead of a zero output, produces a very small value) or to initialize weights with higher values.

The input of the CNN models can be 1) raw waveforms [12, 66], where the data can be processed at a sample-level or at a frame-level or 2) spectrogram-like inputs. SoundNet is trained to use raw audio files in a sample-level way, were the filter size in the bottom layer may go down to several samples long. For the frame-level approach, the first layers compute a hopping window with 100% or 50% hop size, what is known as stride convolution. This layer is expected to learn a filter-bank representation similar to the filter kernels in a time-frequency representation, having the output the same dimension as the mel-spectogram.

In general, audio is of variable-length, therefore CNN have to adapt such an input to a fixed output-size. Different approaches can be used, divided into two main groups, variable-length input or fixed-length input. If the input data is fixed-length, then the common approach is to define a final fully-connected layer, with a global pooling aggregation scheme before [152]. If considering variable-length input, which is more convenient when there are considerable differences in length within intra- and inter- class audio events, there are three common approaches for the final layers of the net:

- **Temporal pooling**, the last layer performs a subsampling procedure using a simple mathematical operation such as maximum, averaging, etc.
- **Attention methods**, giving weighted latent representations to what is important.
- **Recurrent Neural Networks (RNN)**, used to summarize a given sequence, since they are able to accumulate the observations over temporal or spatial dimensions as state variables.

Recurrent architectures, as shown in figure 2.19, connect the output of a hidden-layer neuron with the input of the same neuron. These networks had the vanishing gradient problem, when gradients are too large or too small, which was solved by introducing Long Short-Term Memory (LSTM) [59]. LSTM networks can retain information across many time-steps using

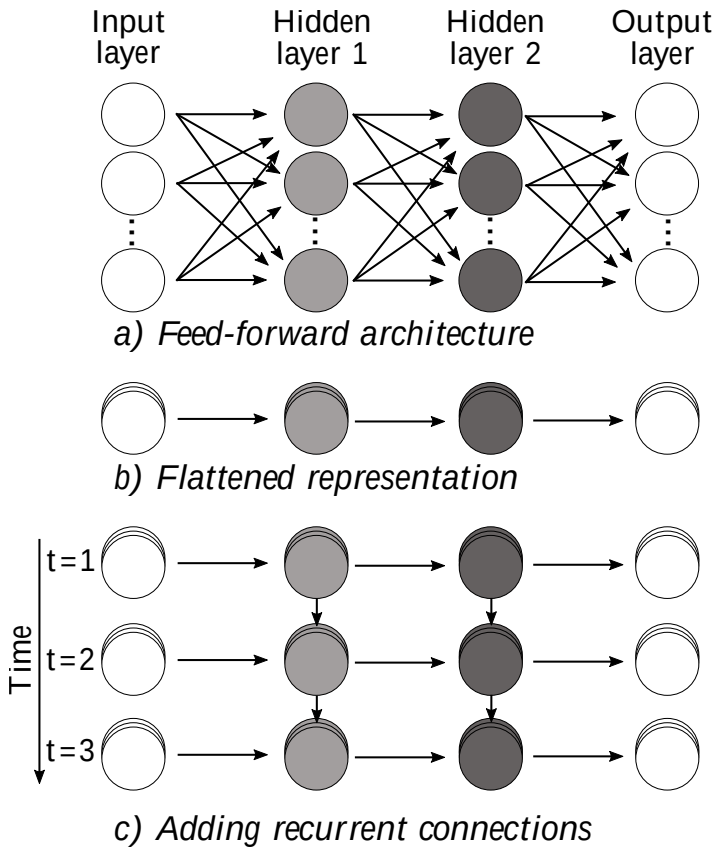



Figure 2.19: Representation of a feed-forward multilayer network with recurrent connections.

memory cells, helping the gradients to be updated through long temporal sequences. The gradient, which depends on an entire sequence, is updated using back-propagation through time (BPTT) algorithm. In this approach the full gradient is approximated by estimating gradients over a finite number of time steps. A simplification of LSTM cells are the Gated Recurrent Units (GRUs), they are also trained using BPTT and avoid as well the vanishing gradient problem. The combination of CNN with a recurrent layer at the end results in a hybrid architecture called CRNN, successfully applied for SED tasks [2, 41].

### End-to-end systems

Given the ability of CNNs to learn internal representations optimized for a given classification task, recent works have proposed the use of raw audio signals as direct inputs to the network [83], avoiding the need to manually extract audio features as a pre-processing step, coined as *end-to-end*. As frequently happens in the machine learning field, end-to-end networks appeared first for solving image classification tasks [72], and were later extended to fields like speech recognition [148, 124], music analysis [35] and environmental sound classification [144]. As evidenced by these works, end-to-end networks can successfully discover frequency decompositions and phase/translation-invariant feature representations from raw audio, making audio analysis systems independent from the selected time-frequency representation and their associated parameters [104].

### Transfer learning

To train CNNs in a supervised way, large amounts of labeled training data are needed. As mentioned in section 5.3.2, there are not large enough datasets with strong labels for the environmental sound domain. The concept of transfer learning is intended to overcome the data scarcity in one domain by seizing the data availability in a different domain. There are different approaches to transfer learning such as, *instance transfer* were the labeled data is re-weighted in the source domain to use it later in the target domain. In *feature-representation transfer*, the goal is to find a good representation that reduces the classification error, while in *parameter transfer* the idea is to find shared parameters or priors between domains. Finally, *relational-knowledge transfer* is aimed at building a mapping of relational knowledge between two domains.

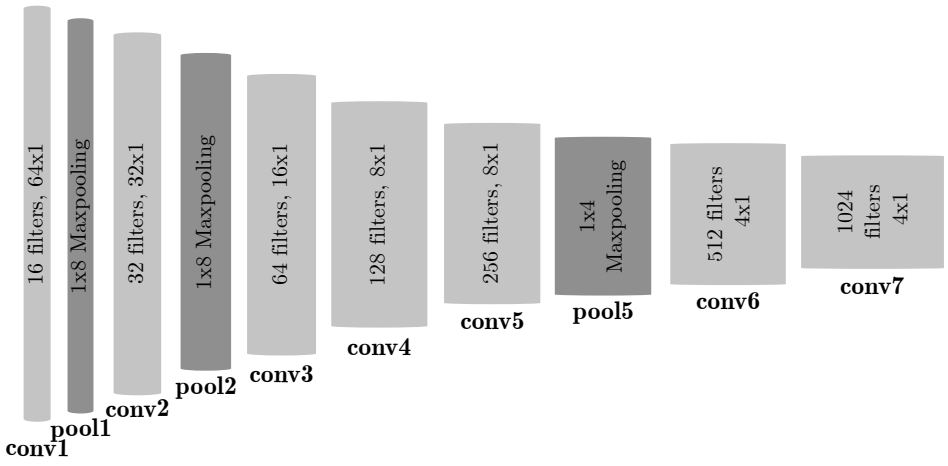



Figure 2.20: Architecture of SoundNet network.

One example of transfer learning between image domain and audio domain is applied in the development of SoundNet model [12]. SoundNet is a fully CNN model with pooling layers in between, the architecture can be seen in figure 2.20. SoundNet was originally trained by using a huge amount of videos from Flickr, following a visual supervision approach where image classification systems (Imagenet, Places) were used to teach recognition models for sound scenes and events (1000 sound classes and 401 scenes). In fact, the SoundNet model was conceived as well to be used as a sophisticated feature extractor based on the useful sound representations learned by its inner layers [96, 95]. This way, the model can be used to classify sound categories not seen during training with a limited amount of training data. Such approach, known as transfer learning [109], has been widely used in the image domain. In the audio processing field it has also been shown to be effective, for example to address the challenging problem of weak labels [37] or for recognizing acoustic events in real-life scenarios [8].

For the experiments carried out during this thesis, the obtained internal representations of the SoundNet CNN model are fed to a linear SVM aimed at performing sound event classification, obtaining close to the state of the art results, following the *feature-representation transfer* approach. We use soundNet model as a feature extractor in Chapter 4, using the data from the layer *pool5*, that can be seen in figure 2.20. The model is also used in further experiments (see chapter 5) as a classifier, adding extra layers at the end to adapt to the given task. To handle different length among the training and testing audio examples, a global pooling layer or LSTM are added at the end, before the softmax activation function.

## 2.8 Open problems

### Data variability

It has been shown in previous sections that environmental sounds present different characteristics than speech or music signals. Sound events comprise a wide range of audio signals having very different spectral and temporal characteristics. Unfortunately, while having great variability among examples pertaining to different classes (inter-class variability) could make easier sound recognition tasks, such variability is also found among examples of the same class (intra-class variability). In fact, sound events may vary also a lot in terms of duration, ranging from few milliseconds up to tens of seconds.

### Scarcity of relevant examples

The number of categories a classifier is able to identify depends on the amount of training data available to train the model. There will be very common sounds from which finding samples is easy and affordable like car passing by, baby crying or bell ringing. However, there some sound events that depending on the context are quite rare.

### Scarcity of annotated data

The scarcity of annotated audio data for training is a major problem when dealing with supervised learning models. Current strong labeled databases contain isolated sound events or synthetically created mixtures in order to facilitate the time-consuming annotation process. These mixtures are synthetic combinations of isolated sounds mixed with some background noises to resemble real-life acoustic conditions.

### Background noise

Background noise is one of the factors that make SER tasks even more challenging. Ambient or background sounds can be caused by weather conditions such as wind or rain. These noises can have different degrees of intensity and contain multiple noisy sources, such as the recordings extracted from a crowded scenario in a concert or at the platform of a train station. Another source of noise is the one caused by the devices used to record the audio examples. The quality of the sound is affected by the microphones attached to portable devices and, in most cases, the results are not the same as when using expensive professional equipment.

### Weakly segmented data

Events appear in every-day situations within a continuous stream of sound. However, when we want machines to process such audio streams, the provided data must have a limited duration with a fixed length. Available audio examples may contain events which are only weakly segmented [46], meaning that the boundaries of the example are not tightly adjusted to the event onset and offset. For carefully built datasets, the events may likely be accurately segmented or temporally annotated within a longer audio stream. Therefore, the system can be trained with data representing more faithfully the content attributable to a given class. In contrast, in realistic applications, the input data entering the classifier is not manually supervised, containing segmentation errors that diminish the performance. Although automatic segmentation techniques can be applied, their effectiveness has to be previously addressed.

#### Poliphony

Real-life acoustic scenes are very complex and they may contain multiple overlapping sounds involving a wide range of categories. While humans are very good at separating a mixture of sound signals and identifying the sources they belong to, machine listening systems may not be prepared to understand such sound mixtures, resulting in a performance poorer than when isolated sounds are observed.

### Mismatched conditions

An added difficulty to the vastness of categories present in sound events is how to accomplish generalization. We want to design learning models able to perform well under changing environments and conditions without the need for manually re-tuning such models. In order to solve this problem, we can either collect large amounts of training data able to cover the wide range of conditions or improve the math. Choosing the second option means to find robust acoustic features against adverse acoustic conditions, like the spectrogram image feature (SIF) [32] successfully applied to convolutional neural networks [168] under noisy conditions. The design of algorithms capable of avoiding overfitting is a challenging task. Current approaches use data augmentation techniques, such as pitch shifting or time stretching, or use convolution operations with distinct impulse responses to simulate different microphone recording and room responses. In deep neural networks, regularization is introduced by adding a dropout layer [137], which works by temporarily and randomly deactivating hidden neurons to avoid overfitting by forcing the network to produce alternative recognition paths.

### Few samples: few-shot learning

As previously discussed, annotating data with strong labels is an expensive task and can take a long time. On the other hand, sometimes it is not possible to collect a large enough dataset for training the supervised model. This situation can occur when the data is very scarce because is too rare (e.g. very unusual animal species) or the task is too specific, such as the identification of infrequent behaviours. In this context, the design of a model able to quickly accommodate the unseen examples with only a very small set of training examples leads to a research area known as few-shot learning. This approach has been investigated in neighbouring domains like image for handwritten character recognition [79] and there have also been some attempts in natural language processing [157]. However, this approach is quite unexplored in the domain of environmental sound recognition. Recently, it has been studied how training with few samples affects different architectures [118], observing that transfer learning models seem to be a promising solution to the problem.

### Open set problem

When all the test examples of a certain application scenario belong to a closed set of categories, it is always known that the observed example must belong to some of the classes present in such set. This is known to be a closed-set scenario. The problem comes when at test time, examples from categories not seen during training appear. In those cases, the example does not belong to any of the predefined classes and then the classifier continues to predict the label from the class that resembles the most. In this scenario, what we want is the classifier to assign an out-of-class label to such unexpected example. By rejecting the example, the classification performance relative to the classes belonging to the set does not get degraded. This problem is known as open-set scenario. To solve this problem, an evaluation metric considering the unknown (out-of-class) category has to be defined. Similar to the

previously mentioned problems, the open-set problem has been addressed first by algorithms in computer vision [132]. The concept of openness in the audio domain was tackled with the use of support vector data description classifiers in ASC [13].

# Chapter 3

# Feature Sensitivity and Robustness Analysis

The way audio signals are analyzed and presented to a given sound event recognition system affects considerably its performance. Feature extraction is a very important step and, therefore, it has to be carefully addressed. Deciding which kind of features or combination of them are selected for a given classification task may already be a difficult problem, with a difficulty that can vary depending on the expertise of the researcher. If the researcher has enough knowledge of the task in question, then he/she may approximate an adequate feature set based on previous experiments. However, even with previous experience, it may happen that the dataset conditions are too challenging for the task at hand, which would likely make a selected set of features perform worse than expected. Traditionally, these hand-crafted features have been divided into time-domain and time-frequency domain features, with varying popularity depending on the final application. Recently, given the limitations of the traditional features and the improvements in generalization achieved by deep models, new approaches for automatically finding good representations have emerged. These techniques are based on the capabilities of convolutional networks for extracting relevant information by only having the raw audio signal or its spectrogram as input.

In the first part of this chapter, a feature selection algorithm is implemented in order to study the importance of each individual feature from a chosen feature set. A ranking of the most important hand-crafted features will be presented, analyzing the results for a particular scenario where there is no mismatch between training and test conditions. The baseline recognition system used for such ranking is then challenged in a simulated realistic scenario, where the test examples are obtained by microphones located inside a room to analyze the performance under varying conditions of noise and reverberation. Similarly, a set of deep features extracted from the SoundNet deep learning framework are analyzed in terms of robustness to the above effects. The results show that in spite of the good generalization capability of deep features, they get affected by noise and reverberation conditions as much as the hand-crafted ones. The outcome of this study motivates the design of further processing techniques for improving the robustness of audio classification features.

## 3.1 Ranking of hand-crafted features

Feature selection is a careful hands-on process that relies on the expertise and knowledge of the researcher. As already explained, there are several kinds of features which, depending on the application domain, can be more useful or less-informative. For a problem like SER, where the sound characteristics among classes are highly variable, one could think that the best solution is to extract a large amount of features and stack them all together. However, by doing this, the complexity of the problem increases considerably, while adding redundant information. To avoid such unnecessary increase of complexity, smaller sets of features have been studied. Given a defined set of features, methods for dimensionality reduction such as PCA (Principal Component Analysis) or ICA (Independent Component Analysis), have been extensively applied in many fields in order to reduce the non-informative data and increase the performance of the model [42]. Note that these methods do not guarantee removing noisy features, they only remove low variance features or separate linearly mixed independent components. However, noisy features show often great variance and, therefore, they are likely to be kept as informative data.

To overcome the ambiguity among feature importance, selection paradigms have been implemented. Obtaining a good feature ordering or ranking is a common and very interesting problem in machine learning, as one may then select an appropriate reduced representation of the data for a given task. Classical feature selection algorithms [63] can be used for this purpose. Selection algorithms are aimed at selecting a subset of features from a larger set of candidates aimed at reducing the classification loss. Even though the task can be seen quite simple, its complexity increases as the set of feature candidates increases. Even the simple and fast sequential family of algorithms still require a quadratic number of model training and evaluations in order to obtain the corresponding feature ranking. In general, the number of subset combinations is extremely high, and also, the classification loss is expensive to calculate for each feature subset. For these reasons, the brute-force search is discarded. Instead, an automatic feature selection algorithm before training the classifier is preferred.

In the literature, different feature selection algorithms can be found, such as SFS (Sequential Forward Selection) and SBS (Sequential Backward Selection), where a feature ordering is obtained by sequentially adding or discarding features in a greedy way. In the particular case of SVMs, works as [34] compared different selection methods in terms of computation cost and efficiency. The results showed that a Sensitivity Based Pruning (SBP) algorithm [149] obtained similar performance results than the iterative SBP or the SBS/SFS sequential selection, with less computation cost. The SBP algorithm was motivated by NN models, where training a quadratic number of models is prohibitive.

In this section, we will focus on studying the sensitivity of common hand-crafted features using the SBP algorithm. The method, the baseline system and the selected set of features considered for such study are presented in the next subsections.

### 3.1.1 Baseline system

The baseline system considered in this section consists of a feature extraction stage and a classification stage. While some approaches include other pre-processing stages to increase the robustness of the system, such as a noise reduction algorithm, we preferred to avoid the use of specific algorithms to concentrate on classifier behavior and feature extraction. The particularities of the system are next described.

### Feature set

The following feature set has been considered to be representative of common hand-crafted features used in general sound recognition tasks. As presented in a previous section, many well-known features were originally conceived within the speech recognition domain. For example, MFCCs or MFEs are still widely used for a variety of audio processing tasks.

**Short-term feature extraction** In the proposed baseline system, features are extracted following a short-term temporal analysis considering Hanning windows of duration 25 ms and 10 ms overlap, with a sampling frequency of 44.1 kHz. For each temporal window, the following features are extracted (in parentheses their number):

1. MFCCs (20): We extract 21 MFCCs, discarding the first DCT coefficient. The considered analysis bandwidth goes from 64 Hz to 8 kHz, with a Mel-scale filterbank consisting of 31 triangular filters. A pre-emphasis filter with coefficient 0.97 is applied to the signal prior to its frequency analysis.

2. Mel Filterbank Energies (MFEs) (31): The energies from the above Mel-scale filterbank are kept as additional features (before applying the usual DCT decorrelation of MFCCs).

3. ΔMFCCs (20): In general, Δ coefficients are calculated as:

$$d_t = \frac{\sum_{n=1}^{N} n(c_{t+n} - c_{t-n})}{2\sum_{n=1}^{N} n^2}, \tag{3.1}$$

   where $d_t$ is a delta coefficient, from window $t$ computed in terms of the static coefficients $c_{t+N}$ to $c_{t-N}$, with $N = 2$. In the case of $d_t = \Delta$MFCCs, the coefficients $c_t$ are any of the computed MFCCs.

4. ΔMFEs (31): As in the above case, when $d_t = \Delta$MFEs, the different $c_t$ coefficients correspond to the computed MFEs.

**Mid-term feature statistics** As it has been previously addressed, in order to obtain a common fixed length representation for all the audio files of different duration, statistics (mean and standard deviation) are extracted from three uniform sections $T \in \{1, 2, 3\}$ corresponding to the onset, midset and offset of the events. The resulting feature vectors are of dimension $(20 \times 2 + 31 \times 2) \times 3 \times 2 = 612$ for each event example.

### Dataset

The algorithm is evaluated using the SED-OL dataset, previously described in Section 2.6. The training data is left as the original, however, for the testing set, we have used a modification of the given development set. The modification is based on segmenting the three given scripted files (of increasing difficulty), using the ground-truth annotations for the individual events. The result are three sets, denoted as *test1*, *test2* and *test3*, corresponding to the three continuous mixtures provided with the original dataset, making a total of 95 isolated audio examples for testing.

### Classifier

The multi-class classifier consists of 16 binary SVM classifiers trained by following a one-versus-all approach, where each binary classifier is trained by considering examples of a

target class as positive data examples and examples from all the other classes as negative data examples. Each binary classifier applies a radial basis function (RBF) kernel with the same parameter, $\gamma = 25$, that was selected as a good tradeoff according to preliminary experiments. To avoid overfitting, a soft-margin parameter $C$ is allowed for each binary classifier, selected after a greedy search procedure in the range $C \in [0.01, 1000]$. Instead of using a validation set, the total accuracy on the same data used to train the SVM is measured along with the number of support vectors inside the margin. The first value that leads to an accuracy above 97% and strictly below 100% is then finally selected.

To classify an individual event, the corresponding features are fed into the 16 trained binary classifiers, obtaining 16 posterior probabilities corresponding to the target (positive) classes. The winning class is selected as the one corresponding to the highest posterior. The extracted features are normalized to have zero mean and unit variance before plugging them into the classifier. Despite its simplicity, this classification scheme gives very reasonable results compared to other state-of-the-art approaches and has been proven appropriate for our feature analysis goal.

### 3.1.2 Feature ranking algorithm

Let $g(\mathbf{x})$ be the function/classifier and let $\mathbf{x} = [x_1, \ldots, x_d]^T$ be its corresponding input feature vector. Let $M_g$ be a particular performance measure on $g$ using a given training set. Let also $M_g$ be a particular performance measure on $g$ using a given training set. The sensitivity of the $i$th feature/input is then given by

$$S_M(i) = M_g - M_{g_i}, \tag{3.2}$$

where $g_i(\mathbf{x}) = g([x_1, \ldots, \bar{x_i}, \ldots, x_d]^T)$ and $\bar{x_i}$ is the average value of the $i$-th input on the training set.

By looking at Eq. (3.2), it is observed that the sensitivity of a given feature (feature $i$), is obtained by computing the performance change on the training set when such feature is replaced by its average in all the examples of the set. This replacement provides a simple way to cancel any discrimination capability attributed to such feature.

Two performance measures are considered as sensitivity criteria:

- $M_A$: the cross-validation (3-fold) estimate of the classification accuracy.
- $M_P$: the average positive posterior probability of each sample corresponding to its true class.

The first criterion corresponds exactly to the one that will be used later to assess the final performance on test data, but it offers a very poor resolution as there are very few training events available. On the other hand, the posterior-based criterion is only loosely related to the final performance (higher positive posteriors do not necessarily guarantee the highest one among all class SVMs) but it allows a finer feature assessment when used as a feature selection criterion. According to these criteria, the sensitivity derived from each one will be denoted as $S_A$ (for the accuracy criterion) and $S_P$ (for the posterior probability criterion).

#### Block-wise feature ranking

A slight variation of the described feature ranking procedure is obtained by grouping the features into meaningful blocks and computing the sensitivity measures for each group as a

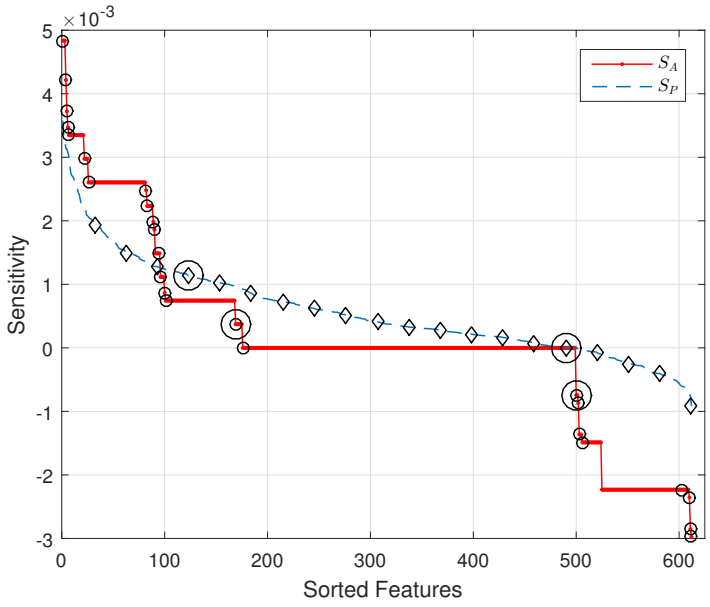



Figure 3.1: Sensitivity values in decreasing order for the two sensitivity criteria considered.

whole. The groups aggregate features from the same family and temporal sections, resulting in 24 different blocks:

- 6 MFE, and 6 ΔMFE blocks, containing 31 features each.
- 6 MFCC, and 6 ΔMFCC blocks, containing 20 features each.

Each of the above 6 blocks, correspond to 3 mean values and their corresponding standard deviations for $T = 1, 2, 3$, respectively.

### 3.1.3 Experiments

The experimentation carried out to evaluate the sensitivity of hand-crafted features has been conducted using standard algorithms for training all SVM models. Feature rankings have been obtained using training data only. Although a 3-fold cross-validation scheme has been used to obtain averaged sensitivity measures, the whole process has been repeated 4 times using different random partitions over the training data. Parameter tuning has been done in each fold in an automatized way as explained in Section 3.1.1 and measures correspond to the average over the four times three folds.

#### Sensitivity analysis

**Feature-wise results** The resulting sensitivity values using classification accuracy, $S_A$, and averaged posteriors, $S_P$, are shown in Figure 3.1. As it can be observed, $S_A$, takes only a few different values, while $S_P$ exhibits a smoother decrease. On the other hand, both criteria lead to a slow varying region towards the middle of the range compared to the higher variation at both ends. In order to carry out further evaluation, a set of 25 and 21 feature subset sizes have been selected in the curves of $S_A$ and $S_P$, respectively. In the first case these roughly correspond to the smallest sizes for a same sensitivity value and are shown in Figure 3.1 as small circles. In the second one, the 21 sizes (shown as small diamonds) have been taken uniformly along the whole range of possible sizes. Two specific sizes for each curve have been marked with large circles for further discussion in Section 3.1.3.

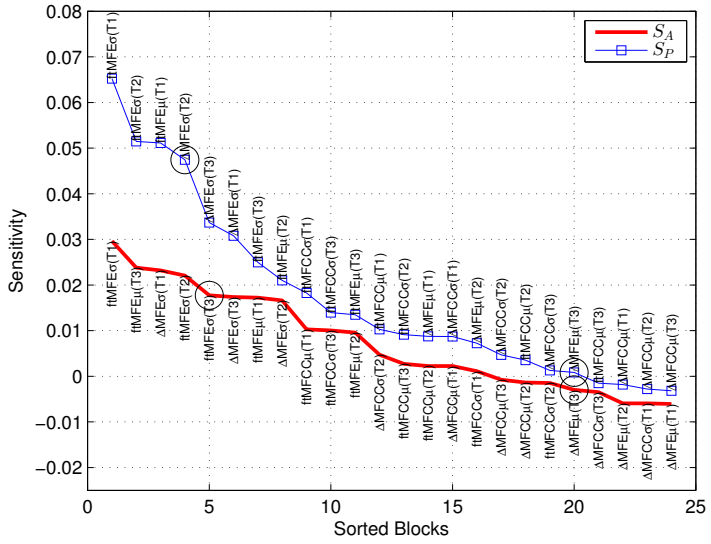



Figure 3.2: Sensitivity values per block in decreasing order for the two sensitivity criteria considered.

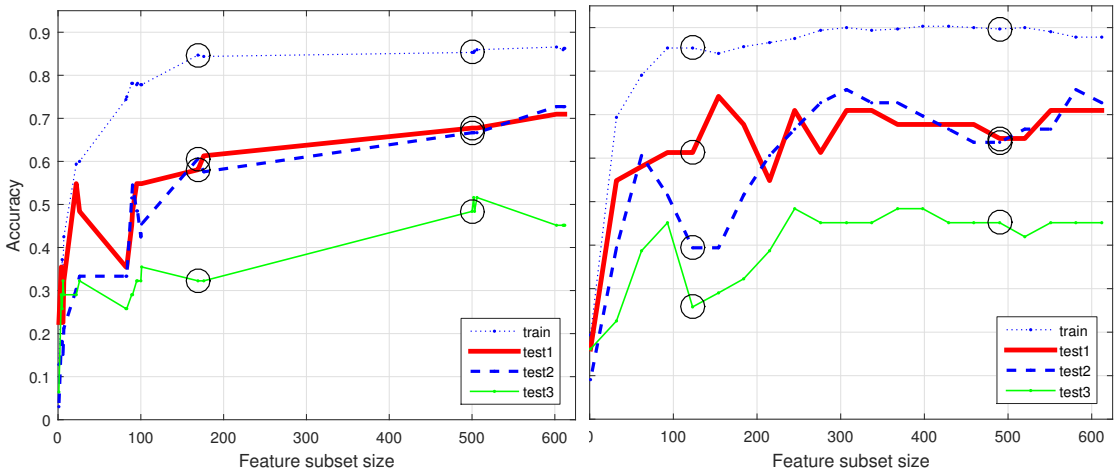



Figure 3.3: Performance obtained when using subset sequence when features are ranked using (a) $S_A$ and (b) $S_P$. Curves show CV estimates using training data (train) and test accuracy (test1, test2, test3).

**Block-wise results** Figure 3.2 shows the sensitivity when considering blocks of features using both the $S_A$ and $S_P$ criteria. The curves now show all 24 different values, ordered from higher to lower sensitivity. In this case, labels are shown to indicate the specific block order. The block results for $S_P$ show higher sensitivity than in the $S_A$ case, specially for the blocks ranked first (from 1 to 5). Interestingly, both criteria tend to show that the performance is more sensitive to MFE features than to MFCC features. In any case, further experiments will be later presented to elaborate on this aspect.

### Accuracy vs dimensionality

After sensitivity analysis, the goal is now to show how classification performance degrades as dimensionality is decreased along the corresponding sequences of nested feature subsets identified by each criterion. To this end, two different learning and classification experiments

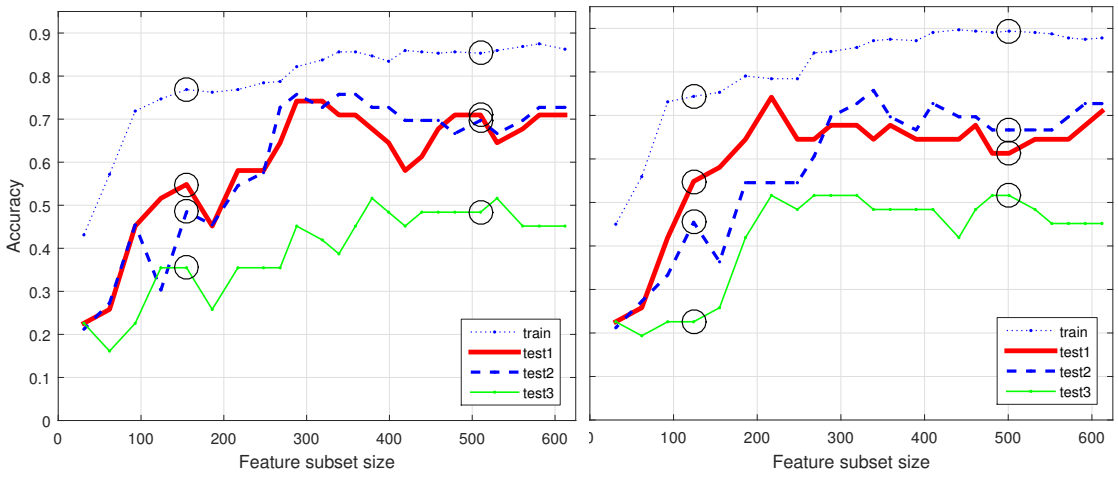
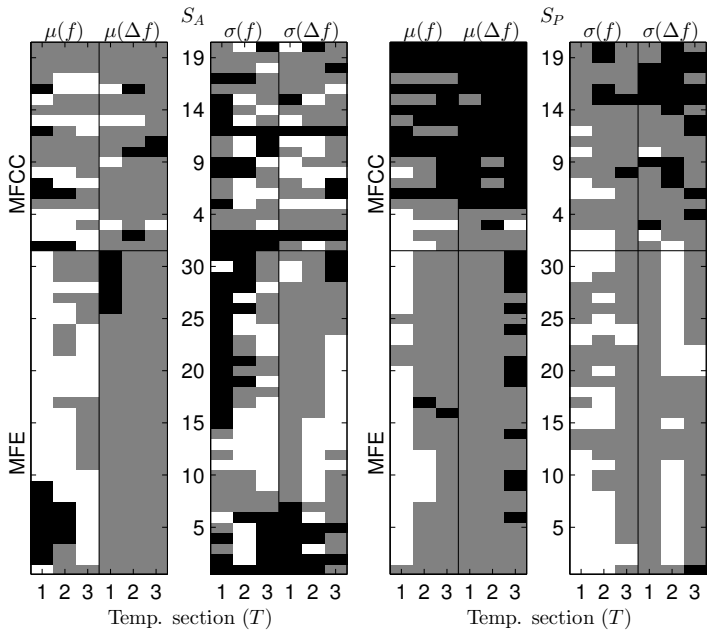



Figure 3.4: Performance obtained when using subset sequence when blocks are ranked using (a) $S_A$ and (b) $S_P$. Curves show CV estimates using training data (train) and test accuracy (*test*1, *test*2, *test*3).

have been designed. On one hand, 3-fold CV-estimates of the accuracy have been obtained using training data. On the other hand, final SVM models have been constructed using the whole training dataset and have been used to classify the 3 test datasets available. All these classification results are shown in Figure 3.3 (a) and (b) for each subset sequence, respectively.

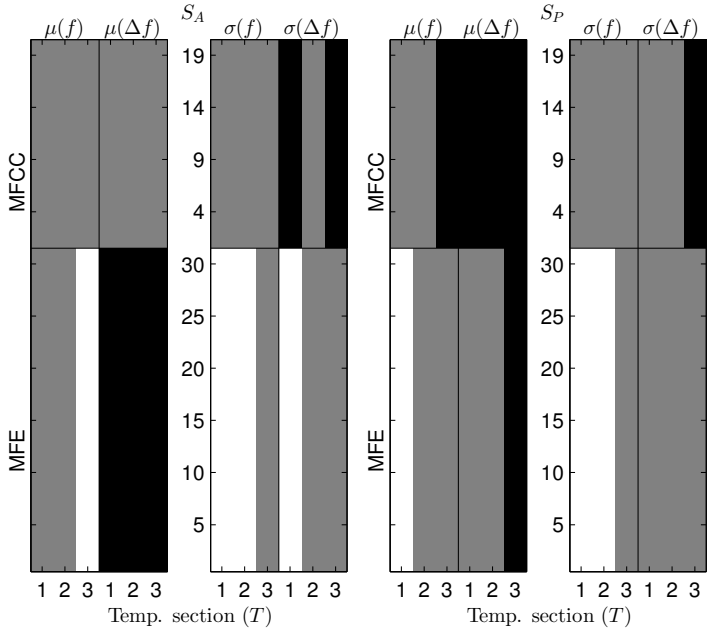
As expected, CV accuracy estimates show a consistent and smooth behavior as dimensionality is increased in both figures when considering the training dataset. In both cases, for this dataset, the accuracy tends to increase until reaching a subset size of 150 ranked features. From this size up to a subset size of 500, the performance gets stabilized for both criteria. However, while similar trends can be observed for the test datasets, the accuracy curves show a more varying structure, especially in the case of $S_P$. Note that the results for *test*1 and *test*2 are similar in both cases, but *test*3 shows a significant lower accuracy. For $S_P$, the *test*2 and *test*3 datasets also show a deep decrease around a subset size of 100 that recovers after 200.

The results of the same experiment but using blocks of features are shown in Figures 3.4(a) and 3.4(b). As with the case of individual features, the performance tends to increase, but at a lower rate. This is due to the fact that the resulting feature subsets are obtained by aggregating features of the same family, where some of them might not be interesting at all from a sensitivity point of view. Also, the training set results are smoother than the ones from the test datasets where, again, similar curves are obtained for *test*1 and *test*2 and worse performance for *test*3.




(a) (b)

Figure 3.5: Feature subsets selected using (a) $S_A$ at sizes 169 and 460, and (b) $S_P$ at sizes 134 and 491. Features in white/black are the most/least sensitive ones and features in gray shade correspond to sensitive values in between.




(a) (b)

Figure 3.6: Feature subsets selected using (a) $S_A$ at sizes 155 and 510, and (b) $S_P$ at sizes 124 and 490. Blocks in white/black are the most/least sensitive ones and Blocks in gray shade correspond to sensitive values in between.

**Feature sensitivity maps**

Two representative feature subset sizes have been chosen and are marked with big circles on the different curves in Figures 3.1 to 3.4(b). To graphically see which are the most important features according to each criteria, feature maps displaying three different importance levels have been produced. In particular, in Figure 3.5(a), the best 169 features according to $S_A$ are shown in white. The next best ones that sum up to 500 are shown in gray. And the remaining ones up to 612 are shown in black. Exactly the same is shown in Figure 3.5(b) for sizes 123 and 490 ranked according to $S_P$. Each map consists of two submaps (mean values, $\mu(\cdot)$, and standard deviations, $\sigma(\cdot)$, respectively) organized as follows. Features ($f$) and $\Delta$ features ($\Delta f$) are displayed at the left/right hand side of each submap while MFCCs and MFEs are at the top/bottom both separated by thin black lines. The labels 1, 2 and 3 at the bottom of the maps indicate whether each column refers to the first, second or last temporal section of the event.

Similar maps for feature blocks are shown in Figure 3.6. The subsets chosen for the block analysis are [155, 510] for sensitivity $S_A$ and [124, 490] for sensitivity $S_P$. Note that the subsets are not equal to those used for individual features because the limits must match with the start and end of blocks. Nevertheless, these slight differences do not affect the comparison of the maps obtained by both approaches (individual feature sensitivity and block feature sensitivity).

## Discussion

From the maps in Figure 3.5 and 3.6, it is clear that $\Delta$ features are globally less important than their corresponding features. Nevertheless, it is possible to distinguish some very important among the $\sigma(\Delta f)$. Another global fact somewhat unexpected is that MFE features are in general more important than MFCCs.

Moving into the comparison between features selected by each criterion, we observe that the best hundred features in the case of $S_P$ (Figure 3.5(b)) are better concentrated at MFEs except in the case of $\mu(\Delta f)$. At the same time, worst features are also concentrated at the mean MFCC part. In the case of $S_A$ (Figure. 3.5(a), important features are spread around the whole submaps but still there are more important features at the MFE regions. In fact, when observing the block sensitivity map in Figure 3.6, it can be clearly observed that MFE blocks are also in general more important than MFCC blocks. Only for the means of $\Delta$MFE features the sensitivity tends to be lower than for the corresponding MFCC block. Interestingly, this fact also seems to hold when studying the individual feature sensitivity map. It is also interesting to analyze the sensitivity obtained per temporal section. Features corresponding to the first and middle parts ($T = 1, 2$) tend to be more important considering both the individual and block sensitivity maps. Only in few cases the last section ($T = 3$) seems to be more important. This difference is more evident when using the posterior-based criterion $S_P$.

In summary, while differences exist between sensitivity criteria, it seems that MFEs have globally a stronger impact on the classification performance than MFCCs. Similarly, the importance of $\Delta$ features tends to be lower than their corresponding feature values. In terms of statistics, the relevance of the means and standard deviations seem to be quite balanced, although the ones computed from the last temporal section of the event have clearly a less significant impact on the classification performance. Finally, it is important to remark that special care must be taken in generalizing the results of this experiment, which has only considered a specific (and limited in terms of number of examples) audio dataset extracted

from a given acoustic environment.

## 3.2 Robustness of hand-crafted features

Real-life environmental events are captured under a variety of acoustic conditions. Background noise and reverberation are common factors affecting the performance of most audio signal processing algorithms, not only when dealing with sound recognition tasks but also in other problems such as sound source localization or signal separation. Background noise is usually due to stationary sounds such as air conditioning (e.g. in indoor locations) or traffic noise (in outdoor environments). On the other hand, reverberation is also a common problem in realistic scenarios, where the interference created by room reflections degrades substantially the original signals emitted by a given sound source. In the previous section, a discussion on the importance of different types of hand-crafted features in the overall system performance, has been provided. However, the experiments carried out so far were only performed under controlled conditions, where neither the training nor the testing data had been artificially degraded to simulate adverse conditions. This section addresses the training/testing mismatch problem introduced in Section 2.8, analyzing the influence of noise and reverberation in the overall performance of a sound recognition system.

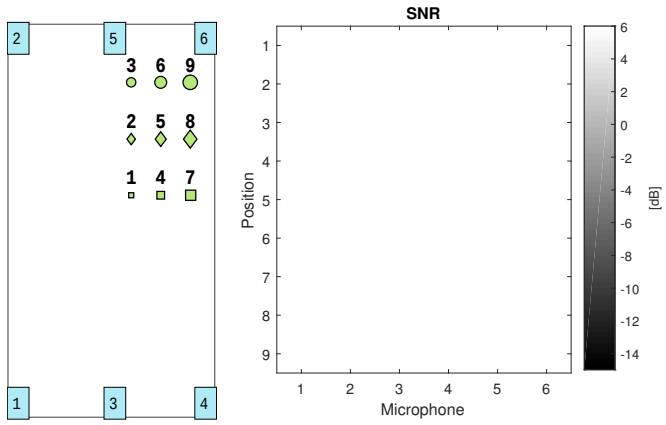
### 3.2.1 Baseline system in simulated adverse conditions

Due to the intrinsic difficulty of SER tasks, most published approaches have been focused on improving the performance of recognition systems considering one of the multiple public datasets available, using subsets for training and testing. However, systems trained with a general dataset may not perform well in a real-life application scenario, even if the sound classes considered are the same.

To address this issue, we evaluate in this section the baseline recognition system described in Section 3.1.1 by simulating a more realistic application scenario, where multiple distributed microphones are available to perform acoustic monitoring within an indoor space. In fact, when multiple microphones are deployed within the monitored space, the diversity offered by the system can be exploited to improve the results provided by a single sensor system. In this case, the use of proper data fusion strategies is yet another point that introduces additional degrees of freedom in the design of SER systems.

#### Simulated scenario

A multi-microphone SER scenario is considered, where 6 microphones are distributed at the front and back walls of a rectangular room (10 m × 4 m × 3 m) at a height of 2 m, as shown in Figure 3.7(a). Taking into account the symmetry of the setup, 9 different positions have been defined within the upper-right quadrant of the room, being representative of the spatial diversity offered by the selected microphone configuration (positions in other quadrants would result in similar cases involving other microphones). Two acoustic scenarios have been considered: anechoic and reverberant, where the last one was simulated using the image-source method [4]. The reverberant scenario had a wall reflection factor of $\rho = 0.8$, resulting in a reverberation time of $T_{60} = 0.31$ s, providing synthetic impulse responses between each position and microphone location. In addition, noisy conditions are simulated by adding background noise to all the simulated microphone signals, fixing an SNR of 6 dB at the microphone closest to the source position. These values allow to simulate a scenario




(a) (b)

Figure 3.7: (a) room spatial map showing microphones (as squared numbers) and event positions (as numbers and symbols). (b) SNR at each microphone for all the defined positions

where at least one of the microphones provides a signal with moderate quality. Figure 3.7(b) shows the SNR at each microphone for each room position. Note that, while the maximum reaches 6 dB at each position, the average SNR at the different positions is different (see Table 3.1).

Table 3.1: Average SNR over all microphones for each position

| Position | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| SNR [dB] | 3.4 | -0.6 | -7.7 | 3.4 | 0.9 | -2.9 | 2.5 | -0.9 | -8.4 |

### Datasets

Taking into account the above simulated scenario, the following datasets are considered to carry out a set of experiments aimed at determining the robustness of the selected set of hand-crafted features in diverse acoustic conditions, considering as well the best case where the information from multiple microphones can be exploited by means of some information fusion strategy.

All the audio examples taking part in the experiments were generated from the SED-SA training dataset, previously described in Section 2.6. This dataset is very similar to the one used for the analysis of sensitivity in this chapter, but the data is augmented to simulate different adverse conditions. As a result, we distinguish among the following augmented testing datasets:

- **Anechoic/noiseless condition** (*clean*): As the name itself says, the test examples are not degraded in any artificial form.
- **Reverberant/noiseless condition** (*rev*): Only room reflections degrade the sound examples.
- **Anechoic/noisy condition** (*6dB*): Only background noise degradations are considered.

- **Reverberant/noisy condition** (*6dBrev*): Both noise and reverberation affect the recordings.

Taking into account the above conditions, different versions of the original sound examples are synthesized for each test condition and each defined room position, generating 6 microphone signals for every original sound example. As a result, 36 different test datasets (4 conditions $\times$ 9 positions) are constructed, where each test example consists of 6 microphone signals.

### Classifier

As in the sensitivity analysis conducted in this chapter, the multi-class classifier consists of 11 binary SVM classifiers trained by following a one-versus-all approach and considering a radial basis function (RBF) kernel. The kernel parameter $\gamma$ and the soft-margin constant $C$ are selected for each binary SVM using grid search and 10-fold cross-validation on training samples only.

A fitting procedure is used to map classification scores to posterior probabilities at each binary machine. The predicted class, $n^*$, is the one corresponding to the maximum output value given the input feature vector:

$$n^* = \arg\max_n \{L_n\}, \tag{3.3}$$

where $L_n = \mathcal{L}_n(\mathbf{x})$ is the posterior probability output obtained from the classifier corresponding to class $n$ given an input feature vector $\mathbf{x}$. The set of hand-crafted features considered to form $\mathbf{x}$ are kept the same as in the feature sensitivity analysis (Section 3.1.1).

### Data fusion

When different predictors are used by using the signals available at each microphone, fusion can be performed either at a decision level or at a probabilistic level. When fusion is performed at a decision level, the class labels obtained by a set of classifiers are considered to decide a final event label. In contrast, probabilistic fusion strategies are based on combining class posterior probabilities before deciding the winning class. Both types of fusion strategies admit weighted modifications where the representations or predictions obtained from each microphone channel are given a confidence or reliability score. Majority voting, maximum posterior rule, product/addition rules or the maximum signal-to-noise ratio (SNR) criterion have already been proposed in this context [143, 51]. Alternatively, if only one predictor is used, data fusion can be performed at a signal level, where the signals/features acquired by the different microphones can be combined prior to classification to reduce the training/test mismatch at the classifier. For example, the use of delay-and-sum beamforming provides enhanced signals that may contribute to make the system more robust to noise and reverberation. The distinction between fusion strategies combining the output of multiple predictors and those merging the information of multiple sources before feeding a single predictor has been referred to as later fusion and early fusion strategies, respectively [10].

Particularly, in this section, several classic late fusion strategies have been considered. Late fusion methods use predictions from each channel $i$ to agree a unique answer (class $n^*$) for each input audio event. In all cases, the weight, $w_i$, for each microphone channel will be assigned in terms of the relative SNR of that channel, i.e.

$$w_i = \frac{10^{\mathrm{SNR_i}/10}}{\sum_i 10^{\mathrm{SNR_i}/10}}. \tag{3.4}$$

**Reliability-based selection (*selec*)** The objective is to select one out of all the available predictions at each microphone channel using a particular reliability criterion or weight:

$$n^* = \arg\max_n \left\{ L_n^{i^*} \right\}, \text{ where } i^* = \arg\max_i \{w_i\}. \tag{3.5}$$

For all the microphones $i$, the one with highest confidence is selected ($i^*$), storing the likelihood scores obtained for all the classes at that specific microphone. The selected class is the one with the highest likelihood score. Note that only one (hopefully the best) of the predictions, the one coming from the $i^*th$ microphone, is effectively used by this method.

**Likelihood accumulation or sum rule (*sum*)** In this case, the combined class likelihood is obtained as a weighted sum of likelihoods from each audio channel.

$$n^* = \arg\max_n \left\{ \sum_i w_i L_n^i \right\}. \tag{3.6}$$

The sum rule when referred to posterior probabilities can be seen as a smoother and better behaved alternative to the product rule that naively assumes channel independence [68].

**Voting (*vote*)** The last considered fusion scheme corresponds to the voting methods, where each trained predictor casts a (possibly weighted) vote for each of the considered classes.

$$n^* = \arg\max_n \left\{ \sum_i w_i \Theta_n^i \right\}, \tag{3.7}$$

$$\Theta_n^i = \begin{cases} 1 & \text{if } n = \arg\max_n \left\{ L_n^i \right\} \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

### 3.2.2 Experiments

This section presents the experiments carried out to evaluate the performance of the baseline SER system considering the already presented test datasets simulating different acoustic scenarios. With the aim of providing more reliable results, the experiments are performed using 5-fold cross-validation. To this end, the training datasets are accordingly partitioned, replacing the testing fold by the corresponding examples selected from the above testing conditions. The process is repeated for each of the different room positions in order to analyze the effect of the spatial location of the event with respect to the selected microphone configuration. The results are always given as classification accuracy, calculated as the percentage of test examples with a correctly predicted label from the whole testing set. Also, note that the values are provided as the average obtained across all folds.

#### Spatial variability

The classification accuracy obtained by the late fusion methods described in the previous section are shown in Figure 3.8, where only the worst acoustic scenario including both noise and reverberation (*6dBrev*) is considered. Light-gray bars represent the average performance over all the tested positions, while markers of different shape represent the accuracy when the sound source producing the event is at one of the defined positions within the room. The worst performances across all positions are represented by dark-gray bars. As it can be observed, the method that provides the smallest spatial variability is *selec*, followed by *vote* and by *sum*. Interestingly, this means that the best performing method is as simple

as selecting the prediction made by the microphone closest to the source (as it is the one maximizing the SNR).

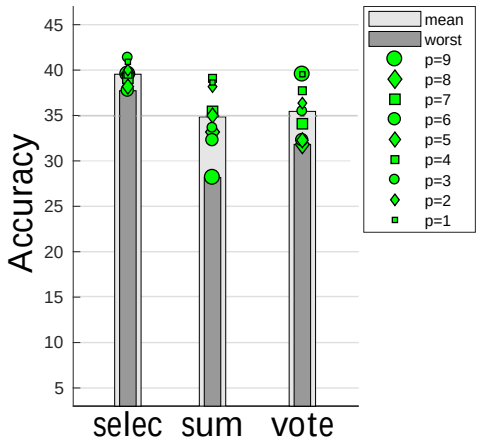



Figure 3.8: Averaged classification accuracy obtained for the late fusion schemes considered.

A spatial map of the performance for the three fusion methods is shown in Figure 3.9, where lighter/darker shades represent better/worse accuracy values. Note that, although the performance of *selec* is equal or better at every position, dependencies with respect to the spatial location of the events are still found in the three cases.

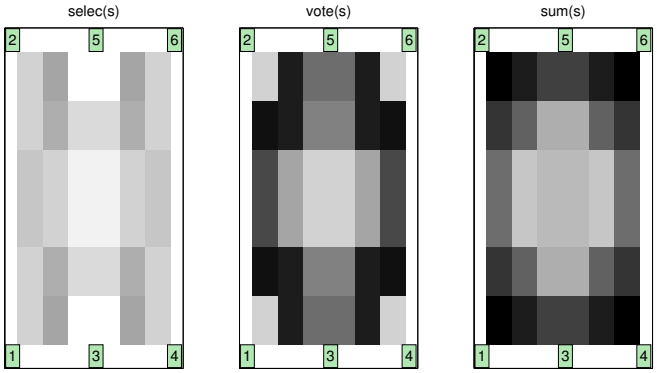



Figure 3.9: Microphone distribution and sound event positions

### Noise and reverberation effects

This subsection analyzes the performance of the three fusion methods as a function of the different degradations considered in the augmented datasets. Figure 3.10 shows how the accuracy decreases for all the methods with respect to the *clean* case where the training conditions match those of the test conditions. Values for the different room positions are indicated by markers of different shape, while lines are used to show their mean spatial performance. As observed for the clean case, no differences are found among the methods when the acoustic conditions match those of the training dataset, achieving a very high classification accuracy in all cases. When only reverberation is present ($rev$), the fusion method achieving best performance is *sum*, followed by *vote* and *selec*. However, when only noise is present ($6dB$) this order is reversed, although all methods experience a significant performance degradation. As

expected, the worst results are obtained for the most adverse acoustic conditions ($6dBrev$), which correspond to the same results represented in Figure 3.8.

In general terms, the accuracy of the system using hand-crafted features gets considerably degraded only with the presence of room reflections, since the accuracy decreases from more than 90% down to 70%. More important is the impact of noise, where the features were observed to be extremely sensitive. In fact, in the noisy case (without reverberation), the performance of the system decreases down to 45%. When both effects are combined, the performance lowers to 35% approximately.

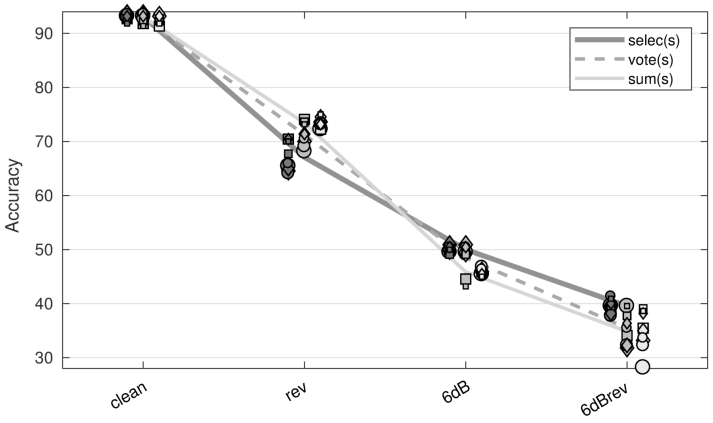



Figure 3.10: Accuracy measure for different acoustic scenarios.

## 3.3 Robustness of deep features

So far, the experiments presented in this chapter have been performed by considering a baseline system based on the classical SER approach, where hand-crafted feature extraction is followed by a fixed-length traditional classifier. Although modern approaches based on deep learning have turned the attention towards convolutional and recurrent neural networks, the classical two-step method can still be very useful in many scenarios, especially those where the amount of available training data is insufficient. Nonetheless, it is a fact that the increase in computational power and available data in the last years, have popularized the use of deep models. The good generalization properties of CNNs in image recognition tasks have motivated the use of such networks as feature extractors. Particularly, CNNs have shown their ability to capture patterns across time and frequency, where each layer reacts to particular aspects of the input that are important to distinguish among samples. This property results in the possibility of suppressing irrelevant information while highlighting the important parts of the data. The use of internal layers as feature representation (known as *deep features*), overcomes the problem of finding a suitable set of hand-crafted features. If deep features are going to be used as classification features, a robustness analysis similar to the one performed for the hand-crafted features is needed.

In this section, we consider degradation artefacts similar to the ones previously analyzed in Section 3.2 for hand-crafted features. However, the baseline system is changed completely to consider a transfer learning framework where deep features learned from a pre-trained end-to-end convolutional neural network are used.

### 3.3.1 Deep learning baseline system in simulated adverse conditions

The baseline system used to study the robustness of deep features is based on the SoundNet architecture, already presented in Section 2.7. As suggested and evaluated by the SoundNet authors, the best performing representations for audio classification are extracted from layer 5 (*pool5*). Similarly, the authors of SoundNet suggest the use of a linear SVM as classification model, following a one-vs-all strategy for solving the multi-class problem. For further details on SoundNet and its recommended use as feature extractor, the reader is referred to [12].

#### Dataset

In a way similar to the analysis performed before in this chapter, adverse acoustic conditions are simulated by augmenting a publicly available dataset. In this case, the selected database is the ESC-50, explained in detail in Section 2.6. The data is split into 5 folds for cross-validation, where the folds for testing are synthetically modified to simulate adverse conditions, while the training folds are left as the originals.

The simulated conditions are similar to the ones already presented for hand-crafted features in Section 3.2.1. The room had the same dimensions and reflective properties, although in this case only a single microphone scenario was considered. The microphone was placed at position 6 (Figure 3.7) at a height of 2 meters, and the sound source location emitting the event was randomly changed among the different positions (from 1 to 9). In this case, different noisy conditions were generated by adding different levels of realistic noise, leading to various Event to Background Ratios (EBRs), EBR $\in \{-6, 0, 6\}$ dB, as defined in [77]:

$$\mathrm{EBR} = 20 \log_{10} \left( \frac{E_{rms}}{B_{rms}} \right), \tag{3.9}$$

where $E_{rms}$ and $B_{rms}$ are the event and background root mean square measures.

Taking into account the above conditions, we generated two test scenarios:

- **Anechoic** (*anec*): the examples of the test folds include different levels of background noise and no segmentation noise is added.
- **Reverberant** (*rev*): besides including noise, the examples of the test folds are convolved with synthetic reverberant impulse responses.

Note, however, that for each of the above scenarios we simulate 4 different EBR conditions, making a total of 8 augmented test datasets.

### 3.3.2 Experiments

This section evaluates the performance of the reference deep learning framework under the varying acoustic conditions discussed above. Table 3.2 shows average Accuracy values when background noise is gradually increased considering both anechoic (*anec*) and reverberant (*rev*) scenarios. Parentheses indicate the standard deviation across the different folds.

First, it is important to notice that the difficulty of the classification problem has increased with respect to the previous experiment. The model has now to classify 50 sound events while before the dataset considered (SED-SA) had only 11 categories. In any case, from the results in Table 3.2, it can be observed that state-of-the-art accuracy can be achieved [114] using a linear SVM trained with deep features. For noiseless conditions (clean), the effect of reverberation causes a drop in Accuracy close to 30% with respect to the anechoic case. Similarly,

| EBR (dB) | clean | 6 | 0 | -6 |
|---|---|---|---|---|
| *anec* | 72.9 (3.4) | 49.8 (3.9) | 35.8 (2.5) | 18.8 (2.1) |
| *rev* | 44.3 (2.9) | 29.9 (2.8) | 22.1 (2.6) | 11.2 (2.1) |

Table 3.2: *Accuracy* performance values for the *ESC-50* dataset under adverse conditions.

for anechoic conditions, the performance drops approximately a 20% for every 6 dB increment in background noise level. As expected, in the reverberant scenario, the performance is always lower than in the anechoic one at every EBR level, with higher differences between both conditions at lower noise levels.

The performance using deep features gets degraded resembling in some aspects the behavior observed with hand-crafted ones. However, while the effect of reverberation is roughly comparable (30% degradation), deep features seem to be more robust to the effect of noise. In this context, a degradation of approximately 25% is observed when only noise is present at EBR = 6 dB, while similar conditions resulted in a 40% degradation when analyzing the robustness of hand-crafted features.

## 3.4 Conclusion

Feature selection is an important aspect of every recognition system. When it comes to the recognition of sound events, classical approaches have traditionally made use of well-known fatures such as the energies from mel-scale filterbanks or mel-frequency cepstral coefficients. In contrast, modern approaches based on transfer learning rely on feature representations extracted from the inner layers of a pre-trained deep neural network, which are generally known as deep features.

In this chapter, we have studied several aspects related to the above sets of features with the main objective of gaining further insight about the generalization capabilities of sound event recognition systems under changing acoustic conditions. To this end, some experiments have been presented that make evident that further work is needed in this regard. First, the importance of a well-known family of hand-crafted features has been analyzed, concluding that while differences exist between sensitivity criteria, MFEs have globally a stronger impact on the classification performance than MFCCs. The same family of hand-crafted features was analyzed in a multi-microphone scenario for different conditions of noise and reverberation without re-training the system. As expected, a relevant degradation in classification performance was observed, even when late fusion techniques are employed. Finally, similar effects were considered for deep features, demonstrating that, despite being slightly more robust to noise, acoustic mismatches have also a strong impact on the performance in deep learning-based frameworks.

# Chapter 4

# Adaptive Mid-Term Representations for Event Classification

In the previous chapter, it was emphasized the importance of feature selection in sound event recognition, with a special focus on hand-crafted features. Apart from selecting a suitable set of features that are representative enough to perform the classification task, these features have to be able to generalize under different acoustic conditions. As it has been shown, both hand-crafted features and deep features, are affected by adverse acoustic conditions such as background noise or reverberation. These conditions are common for real-life environments, where sound events are rarely isolated and most of the time appear in combination with background noise or other sound events. In Section 2.3, the different temporal approaches for SER have been introduced, dividing them into sequential (or online methods) and segmentation-based (or offline methods).

A major drawback of online approaches is the selection of an optimal sliding-window length suited for all the considered classes. Some sound events are short-lived (e.g gun-shot) and others are characterized by a longer duration (e.g. telephone). If the window length is too small, the long-term variations in the signal would not be well captured by the extracted features, and the analysis procedure might divide events into multiple windows. On the other hand, if the window length is too large, it becomes difficult to locate segmental boundaries between consecutive events and there might be multiple sound events in a single frame or the window may contain a high proportion of background noise. Additionally, most offline approaches use a pre-processing step to detect the audio part corresponding to a potential event, isolating it from the continuous audio stream before extracting the desired features. While this seems convenient from a classification point of view, the system becomes also dependent on the performance of this stage, with segmentation errors affecting the classification accuracy.

This chapter presents a mid-term analysis technique suitable for fixed-length input classification robust to inaccuracies in the segmentation stage. The proposed technique can be interpreted as an adaptive texture window approach based on a uniform subsampling of the audio signal in the feature space. This subsampling procedure is performed according to the accumulated distance observed over another feature space capturing the temporal structure

of the event. This results in a fixed-length representation of the signal that inherently encodes its temporal evolution. Thus, texture windows applied over this new representation are actually related to meaningful sections of the event having different temporal lengths. Our approach has shown to improve the classification performance when segmentation errors are present in the test audio clips. The pre-processing step improves the generalization capabilities of the model since it was trained using perfectly segmented sound events.

## 4.1 Related work

The issue of having different-length inputs in audio classification tasks has been considered problematic since the development of early automatic speech recognition (ASR) systems, especially in those focused on recognizing isolated words. Template matching approaches making use of Dynamic Time Warping (DTW) have been used for long in ASR systems, being a widely used method in this task. Aligning speech utterances of different lengths is addressed in DTW by warping the time axis repetitively until an optimal match is found, performing a piecewise linear mapping in the temporal dimension. Note, however, that in order to perform this alignment, the speech utterances must be correctly segmented, resembling the procedure followed by offline AED approaches. The use of DTW for recognizing natural sounds such as bird singing has been proposed in [5]. Also, DTW-based classification kernels have been already proposed in [141], showing that, while DTW may be useful for classifying events having a clear temporal structure, it also leads to some errors when this is not the case.

In [7], the problem of varying input lengths in ASR was addressed by proposing the use of the trajectory matrix outerproduct. In this method, speech utterances are mapped to a fixed-length input by vectorizing the outerproduct matrix of the short-term MFCC features corresponding to a set of temporal sections. It is shown that the performance of models trained with patterns extracted with the outerproduct matrix method is significantly higher than that of linear compaction and elongation [134]. Another popular approach for speaker recognition and verification is the i-vector representation [31], which provides a fixed-length, low-dimensional representation of audio segments by using Factor Analysis (FA) and Baum-Welch statistics. Note that, in all cases, it must be taken into account that mapping a varying-length sequence to a fixed-length one may involve loss of information, introduction of distortion and discontinuity, which will affect the performance of the classification models [7].

More recently, the problem of matching sequential data taking into account its temporal structure has arisen in diverse application environments, such as motion sensors, e-pens, eye-trackers, etc. In [84], a warped k-means method for sequential data clustering is proposed in this context, where trace-segmentation (TS) is used as a preprocessing step. TS has been traditionally employed in ASR and writing recognition to create a resampled sequence, or "trace" by using a piecewise linear interpolation based on distances accumulated along the original data trace. The use of TS was initially proposed in [73] to perform a non-linear time-normalization of a sequence of speech frames in ASR. The method was shown to exceed the performance provided by DTW, reducing as well its computational complexity and memory requirements [23].

The approach presented in this chapter makes also use of TS in order to embed the temporal evolution of the events into fixed-length feature vectors used for classification. To this end, as it will be described in detail, two different feature spaces are considered: one used for trace analysis, which is assumed to capture faithfully the temporal behavior of the event, and the actual feature representation of the event aimed at providing class discrimination.

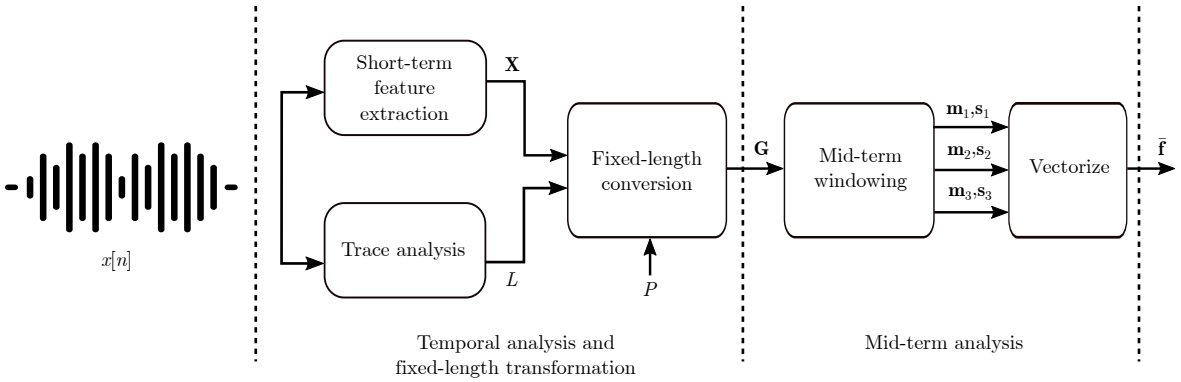



Figure 4.1: Block diagram of the proposed approach.

Finally, it is also important to remark here that, while the present work is highly related to the AED field, its novelty does not reside on the proposal of a new AED system that detects the presence of a given class event within an input audio stream. As it will be presented throughout the following sections, our contribution is fully related to a non-linear feature transformation method aimed at improving the robustness of audio classifiers to segmentation errors, i.e. errors resulting from incorrect onset and offset timestamps.

## 4.2 Proposed approach

This section presents the different processing steps that make up the proposed approach. The block diagram shown in Figure 4.1 differentiates into two different stages. First, audio features are extracted from the input sequence, where a trace analysis block is considered to obtain a fixed-length representation of the event taking into account its temporal evolution. Then, a conventional mid-term windowing is applied over this representation, extracting statistics from uniformly overlapping blocks. The following subsections describe the processing steps included in the aforementioned stages.

### 4.2.1 Short-term feature extraction

Let us consider an input audio signal $x[n] \in \mathbb{R}$ which is assumed to contain an acoustic event. The input signal is split into short-time overlapped frames ($T$) of length $N_w$, extracting a set of $N$ features for each frame. This results in the following feature matrix:

$$\mathbf{X} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_T] \in \mathbb{R}^{N \times T}, \tag{4.1}$$

where $\mathbf{f}_t \in \mathbb{R}^N$, $t = 1, \ldots, T$, are the feature vectors corresponding to the analyzed frames:

$$\mathbf{f}_t = \left[ f_1^{(t)}, f_2^{(t)}, \ldots, f_N^{(t)} \right]^{\mathrm{T}}. \tag{4.2}$$

The individual features extracted from frame $t$ are denoted as $f_k^{(t)}$, $k = 1, \ldots, N$.

The temporal evolution of the event has been classically incorporated into the feature representation by adding the derivatives of the considered features, which are commonly known as $\Delta$ (first derivative) and $\Delta\Delta$ (second derivative) features.

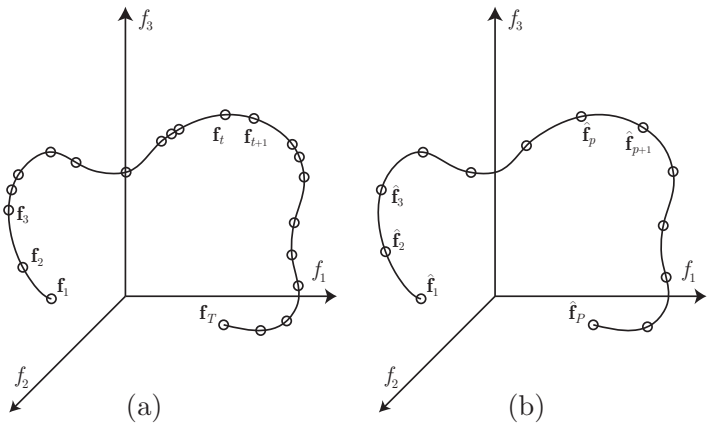



Figure 4.2: Trajectory representation, (a) uniform time sampling (b) uniform distance sampling.

### 4.2.2 Trace analysis

The temporal evolution of the event can be assumed to create a trajectory in the transformed feature space. This idea is represented in Figure 4.2(a), where the different short-term feature vectors are depicted as connected points within an $N$-dimensional space.

A temporal change between two successive feature vectors can be quantified by their Euclidean distance:

$$d\left(\mathbf{f}_t, \mathbf{f}_{t+1}\right) = \left(\sum_{k=1}^{N}\left(f_k^{(t)} - f_k^{(t+1)}\right)^2\right)^{1/2}. \tag{4.3}$$

The cumulative distance at a given time frame $t$ is given by:

$$L(t) = \sum_{i=1}^{t-1} d(\mathbf{f}_i, \mathbf{f}_{i+1}), \tag{4.4}$$

so that the total spatial length of the event trajectory is $L(T)$.

A major problem to appropriately analyze the temporal behavior of the events is related to the particular short-term features used. In particular, features good for discriminating among events can be bad for capturing its temporal structure and vice-versa. More importantly, the dimensionality of the feature space used for trace analysis should be in principle as low as possible to avoid well-known adverse effects [3].

Taking into account the above issue, our proposed approach considers the use of alternative spaces obtained from the time-frequency (T-F) energy distribution. Note that, while the energy envelope of the event might reflect sufficiently well the temporal behavior, frequency changes do also contribute to the perceived temporal structure. For example, consider an event consisting of a sinusoidal signal with constant amplitude but sudden pitch changes. Although its energy remains constant with time, it would be perceived as having a clear temporal structure due to its frequency changes.

Then, to analyze the trace of the event, we construct an alternative to the feature matrix in 4.1 as

$$\tilde{\mathbf{X}} = \left[\tilde{\mathbf{f}}_1, \tilde{\mathbf{f}}_2, \ldots, \tilde{\mathbf{f}}_T\right] \in \mathbb{R}^{K \times T}, \tag{4.5}$$

where $\tilde{\mathbf{f}}_t \in \mathbb{R}^K$, $t = 1, \ldots, T$, are the new feature vectors corresponding to the analyzed frames:

$$\tilde{\mathbf{f}}_t = \left[\tilde{f}_1^{(t)}, \tilde{f}_2^{(t)}, \ldots, \tilde{f}_K^{(t)}\right]^{\mathrm{T}}. \tag{4.6}$$

The following is a list of the T-F representations corresponding to features spaces used for describing the sound event trajectory sorted according to dimensionality and spectral resolution:

- **Power spectrum** (*ps*): Corresponds to the conventional power spectrum of a signal frame, given by:

$$\tilde{\mathbf{f}}_t = \left|\mathrm{DFT}\left\{\mathbf{x}_t\right\}\right|^2, \tag{4.7}$$

  where $\mathrm{DFT}\{\cdot\}$ returns the non-redundant discrete Fourier transform (DFT) coefficients of a signal vector and $\mathbf{x}_t$ denotes the $t$-th temporal frame of the input signal $x[n]$. Note that the dimensionality $K$ is given by half the number of samples in a frame.

- **Subband power** (*sbp*): accumulated power over a set of $K$ logarithmically-spaced frequency bands. We consider $K = 4$ with frequency limits established at frequencies $[0, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1]$, where these values are normalized with respect to the Nyquist frequency.

- **Energy** (*en*): energy of a signal frame computed in the time domain, i.e.

$$\tilde{\mathbf{f}}_t = \sum_{n=1}^{N_w} (x_t[n])^2, \tag{4.8}$$

  where $x_t[n]$ are the samples belonging to the $t$-th frame. In this case, $K = 1$ and frequency information is discarded.

To illustrate their effects in the analysis of event traces, Figure 4.3 shows the normalized accumulated distances (bottom panel) for a given sound event having a well-defined temporal structure (upper panel).

Apart from the above defined T-F features, the $N = 102$ short-term features (described in Section 4.3) and a subset consisting only of the output of a mel-scale filterbank with 31 subbands are shown for comparison purposes. In this plot, profiles close to the diagonal correspond to bad results. On the other hand, T-F features and specially *en* and *sbp*, exhibit high slopes and flat sections that roughly correspond to more and less informative regions along the event, respectively. In addition to *mel* and *N features*, other psychoacoustically-motivated features such as constant-Q transform coefficients and outputs from a gammatone filterbank have also been considered. However, while being computationally more expensive, their use results in traces that only differ slightly from the ones already described.

From Figure 4.3, it should be emphasized that the DFT power spectrum, *ps*, captures the temporal evolution of the event significantly better than *mel* and *N features*, despite having high-dimensionality. This is not a surprising fact, since features used for classification purposes are preferred to be non-correlated and standardized and, while changes may be observed due to their temporal evolution, frame-to-frame distances do not reflect temporal changes as clearly as power-related features.

### 4.2.3 Fixed-length conversion

Once the trace of the event has been analyzed, the next step is to create a resampled version of the feature matrix $\mathbf{X} \in \mathbb{R}^{N \times T}$ into a new one that considers the trace information $L$,

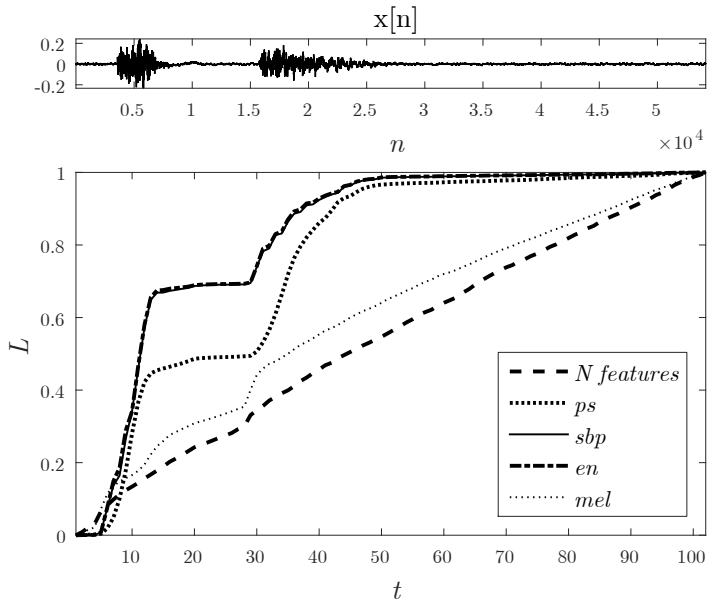



Figure 4.3: Normalized trace ($L(t)/L(T)$) corresponding to the top audio signal (cough class) using different feature sets.

which will be denoted as $\mathbf{G} \in \mathbb{R}^{N \times P}$. Note that $\mathbf{X}$ provides a uniform temporal sampling of the event in the feature space, since its columns are feature vectors $\mathbf{f}_t$ obtained at regular time intervals. The idea is to transform $\mathbf{X}$ into $\mathbf{G}$ by resampling the event feature data with equidistant spatial segments along the event trace. To illustrate this procedure, Figure 4.2 shows how consecutive feature vectors $\mathbf{f}_t$ in $\mathbf{X}$ (a) are not equidistant in the feature space, in contrast to the new points $\hat{\mathbf{f}}_p$ in $\mathbf{G}$ (b). Note that, while $T$ depends on the audio input length, the dimensions of $\mathbf{G}$ are fixed given the number of sampled points $P$.

The new feature vectors are calculated by dividing the total spatial length of the trace, $L(T)$, into $P-1$ uniformly spaced segments. The length of each spatial segment is:

$$L_s = \frac{L(T)}{P-1}, \tag{4.9}$$

where $P$ is the desired number of points. The new feature vectors $\hat{\mathbf{f}}_p$ are estimated by linearly interpolating between the two nearest feature vectors in $\mathbf{X}$. This process is similar to a uniform sampling procedure, where each sample corresponds to points of great variation in the original input sequence.

First, the two vectors of the original feature matrix which lie on either side of the $p$-th segment boundary are found. This is achieved by finding the value of $j$ for which the following inequalities hold:

$$L(j+1) > pL_s, \tag{4.10}$$

$$L(j) < pL_s. \tag{4.11}$$

Then, the feature vectors $\mathbf{f}_j$ and $\mathbf{f}_{j+1}$ are linearly interpolated to find the new feature vector $\hat{\mathbf{f}}_p$ as

$$\hat{\mathbf{f}}_p = \mathbf{f}_j + (\mathbf{f}_{j+1} - \mathbf{f}_j)\alpha, \quad p = 1, \ldots, P, \tag{4.12}$$

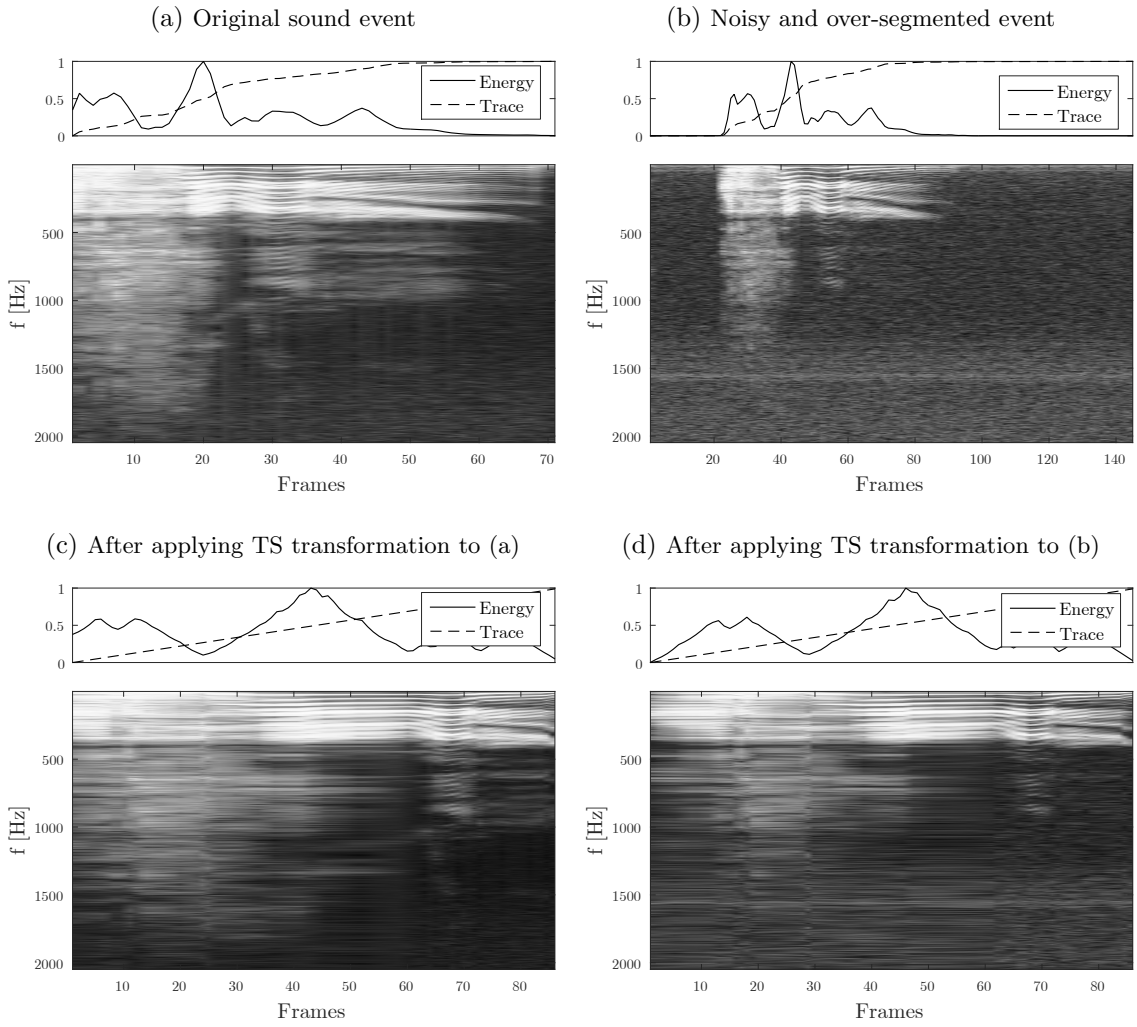



Figure 4.4: Spectrogram of an audio event example, the upper panel shows the trace computed using the accumulated distances of the event energy. (a) is the original audio event, (b) shows a noisy and over-segmented version of it, (c) shows the result of applying trace-based transformation to (a) and (d) is the result of applying the same transformation to (b).

where $\alpha$ is:

$$\alpha = \frac{pL_s - \sum_{t=1}^{j-1} d(\mathbf{f}_t, \mathbf{f}_{t+1})}{d(\mathbf{f}_j, \mathbf{f}_{j+1})}. \tag{4.13}$$

This procedure results in a new set of feature vectors that form the new fixed-length feature matrix $\mathbf{G}$:

$$\mathbf{G} = \left[\hat{\mathbf{f}}_1, \hat{\mathbf{f}}_2, \ldots, \hat{\mathbf{f}}_P\right], \in \mathbb{R}^{N \times P}. \tag{4.14}$$

The new feature matrix $\mathbf{G}$ corresponds to a non-linear time normalization of the input audio sequence based on a distance metric. Note, however, that the transformation takes into account the trace of the event in the reduced T-F feature space rather than the one used for class discrimination, capturing better its temporal evolution.

It is important to note that the value of $P$ can play a very critical role depending on particular event lengths. Low values lead to oversmoothing and information loss while high values may imply highly redundant representations. Therefore, one must carefully choose this parameter as a convenient and relatively robust representative length. For our experiments, $P$ has been set to the median number of frames across the whole database.

To visualize the effects provided by the proposed normalization, Figure 4.4 shows how a given audio event is normalized in length considering the previous steps. For the sake of clarity in the interpretation, the original feature matrix $\mathbf{X}$ is given by the log-magnitude spectrogram of the audio input. The trace $L(t)$ is analyzed taking into account only the frame energy ($en$), being both shown at the top of each example. The figure analyzes two situations corresponding to the same audio event, one where the event does not include noise and is accurately segmented (a) (duration $T = 70$) and another with over-segmentation and noise (b) (with $T = 140$). At the bottom part, the resulting fixed-length conversions of these examples are shown respectively in (c) and (d), using $P = 84$. Note that the result in (d) is very similar the one in (c), which reflects how the conversion procedure has significantly omitted the segmentation imperfections present in (b). Also, the energy changes between consecutive points in the new representation are equal, resulting in linear traces after the fixed-length conversion step.

### 4.2.4 Mid-term analysis

The new feature matrix $\mathbf{G}$ is the input to the mid-term analysis stage, which performs the statistical analysis of the event across a set of uniform texture windows. It must be emphasized that the use of uniform-length windows in the analysis of $\mathbf{G}$ corresponds to a non-uniform temporal analysis of the original audio event, providing a statistical description of each feature along uniform (possibly overlapped) divisions of the trace.

Figure 4.5 indicates schematically how 3 different sections $w_1$, $w_2$, $w_3$, are defined over $\mathbf{G}$. For each feature $f_k$, the mean and standard deviation across each of the defined sections are computed ($\mathbf{m}_i$ and $\mathbf{s}_i$, where $i = 1, 2, 3$). Finally, all these statistics are stacked together and vectorized to produce the final feature vector $\bar{\mathbf{f}}$ used for classification. It is worth to note that applying a mid-term analysis over $\mathbf{G}$ is equivalent to an adaptive statistical description of the event based on its temporal structure. This also reduces the dimensionality of the classifier input while retaining some information on the trace evolution [165, 49].

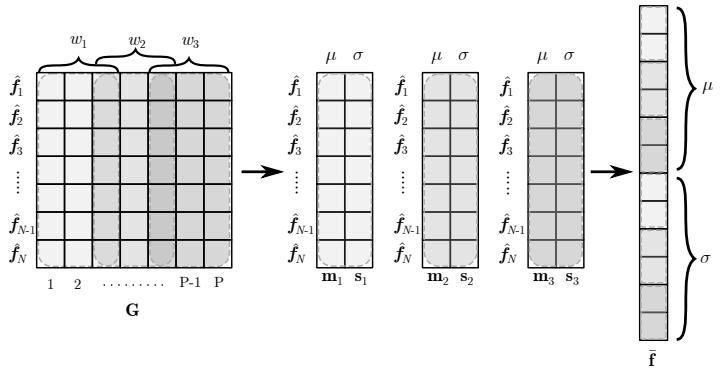



Figure 4.5: Mid-term analysis applied over the fixed-length feature matrix **G**. The statistics are calculated across three overlapped sections represented with $w_1, w_2$ and $w_3$.

## 4.3 Experimental setup

To evaluate the performance of our proposed approach, two different databases were selected, SED-SA dataset and TUT Rare Sound Events dataset, both described in detail in Section 2.6. In order to evaluate the improvements of using our approach under adverse conditions, both datasets have been modified.

For SED-SA the training set is split using a 5-fold cross-validation, where the folds used for testing are modified adding different Event to Background Ratio (EBR) levels, EBR $\in \{-6, 0, 6\}$ dB, defined as in Section 3.3.1. For the testing set, the original development set is modified, segmenting the synthetic mixtures using the provided annotation to obtain isolated examples. From the three monophonic scripts, each containing 3 events per class, we end up with 33 isolated sound events.

The training data for the TUT Rare Sound Events dataset is left unmodified. For the testing set we select only the mixtures containing one of the three classes, which results in 250 examples for each level of EBR. The segmentation from the mixtures is done the same way as in the above dataset. Moreover, this database provides a very challenging test set consisting on real audio scenes in which background noise is non-uniform and may contain artifacts and unwanted events.

With the aim of studying the robustness of the proposed approach to noise and segmentation errors, we extended the SED-SA training set to include more levels of background noise (EBR $\in \{-6, 0, 6, 12, 18\}$) as well as non-active sections before and after the original events to simulate segmentation errors. To this end, realistic noise was added to the event examples for each EBR condition following the procedure described in [76], which is the same used in the DCASE test dataset. Noise examples were freely downloaded from [78].

In the TUT Rare Sound Events dataset, segmentation errors where simulated in the same way but each test event was always surrounded by the particular context from its own scene with a fixed noise level.

Figure 4.4(a) and (b) show an example of the spectrograms corresponding to an original event from the SED-SA dataset and its degraded version, respectively. It should be emphasized again that the classification system is always trained with noiseless samples with perfect segmentation so, ideally, the system is designed to learn from the actual event information (without introducing any prior on possible unwanted effects in the test stage such as imperfect

segmentation or background noise).

### 4.3.1 Trace features and classification features

The baseline recognition system used and the selected features for audio event classification are the same ones already analyzed in Chapter 3. These features are: MFCCs (20), MFEs (31) and their corresponding $\Delta$ features $(20 + 31)$, making a total of $N = 102$ short-term features. We selected these features for being representative acoustic descriptors used in the literature. Note that, while $\Delta$ features may be somewhat redundant when trace analysis is considered, the authors' experiments suggest that the performance gets slightly benefited after their incorporation ($\approx 5$ % on average).

The performance of our proposed approach is analyzed for different trace analysis alternatives, each of them making use of a given T-F feature as described in Section 4.2.2: *subband power (sbp), power spectrum (ps)* and *energy (en)*. For comparison purposes, the performance of a conventional mid-term analysis scheme (without performing trace analysis) is also provided, denoted as *nt*. In our experiments, we selected $P = 84$ trace points, which corresponds to the median number of frames for the individual events in the SED-SA training set. The corresponding length of the mid-term sections $(w_1, w_2, w_3)$ is 36, with an overlap of 1/3 between sections. This overlap value was selected after conducting some preliminary experiments for overlaps ranging from 0% to 50%.

Once the mid-term analysis is performed, the final feature vector consists of $102 \times 3$ mean values and $102 \times 3$ standard deviations, corresponding to the defined mid-term sections for each event example. All these values are aggregated into a final vector of dimension 612. The following subsections describe the details of the classifier and the considered datasets.

### 4.3.2 Classification

The performance of our proposed alternatives is analyzed using two back-end classifiers, namely SVM and Random Forests. The multi-class SVM (*svm*) classifier is trained by following the one-versus-all approach, as in previous Chapter 3. The kernel used for each binary SVM is a radial basis function (RBF), where the $\gamma$ parameter as well as the soft-margin parameter $C$ are obtained using a greedy search algorithm with an internal 10-fold cross-validation. The Random Forest (*rf*) classifier uses 70 trees with a maximum number of features selected as the square root of the total. This was the best performing model after a set of preliminary experiments testing.

For comparison purposes, other classification systems (GMM and HMM) are also compared, both using the same set of features described in Section 4.3.1. The GMM (*gmm*) classifier was trained using the expectation-maximization algorithm and constraining the covariance matrix to be diagonal, as in [135]. We selected the best performing model, which had 32 Gaussian components. The HMM (*hmm*) classifier consisted in a left-to-right fully-connected model where the event activity is modeled by three states [99]. We added also an initial and final state trained on background noise in order to make the classifier more robust to segmentation errors.

Finally, we provide the performance of an state-of-the-art deep-learning-based method, using as features the output of an internal layer of the pre-trained *SoundNet* CNN (*snet*) [12]. Specifically, we use features from the *pool5* layer as suggested by the authors, using

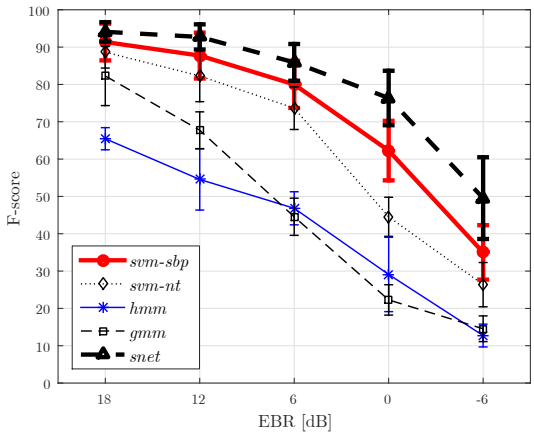



Figure 4.6: Classification performance for perfectly segmented events considering the SED-SA training set under different EBR conditions.

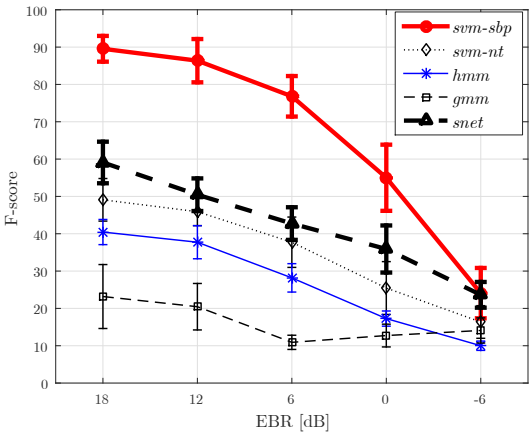



Figure 4.7: Classification performance for wrongly segmented events considering the SED-SA training set under different EBR conditions.

Table 4.1: Reference $F1$ performance values for the systems when clean and perfectly segmented events are considered. The standard deviation is shown in parenthesis, computed over the total number of realizations.

| *svm-sbp* | *svm-ps* | *svm-en* | *svm-nt* | *hmm* | *gmm* | *snet* | *rf-sbp* | *rf-ps* | *rf-en* | *rf-nt* |
|---|---|---|---|---|---|---|---|---|---|---|
| 91.8 | 95.9 | 92.7 | **96.4** | 75.5 | 89.1 | 94.1 | 91.4 | 90.5 | 90.0 | 89.5 |
| (5.9) | (3.0) | (6.3) | (4.7) | (4.4) | (7.1) | (3.4) | (4.9) | (4.1) | (3.4) | (6.7) |

their own code provided at [167].

## 4.4 Results and discussion

### 4.4.1 Robustness to imperfect segmentation and background noise

We first take into account segmentation errors at test time when recognizing sound events in the SED-SA training set. We consider a conservative segmentation procedure where the input audio stream contains completely the event of interest. To simulate this, we add background sections of random length before and after the event that together sum up to the event length. We refer to this situation as 100% segmentation error (see the example in Figure 4.4(b)). For comparison purposes, we also evaluate the performance when perfectly segmented events (0% segmentation error) are considered. Results are computed using a standard 5-fold cross-validation, splitting the dataset into 5 different disjoint subsets of approximately the same size, and using each time one of the subsets for test (with realistic noise added) and the remaining ones for training.

Figures 4.6 and 4.7 show the performance obtained over the SED-SA training set. The events in the training folds are perfectly isolated and do not contain background noise. However, the events in the test fold are modified to match a given EBR condition. Also, the events used for test may be perfectly segmented (Figure 4.6) or wrongly segmented (Figure 4.7) as

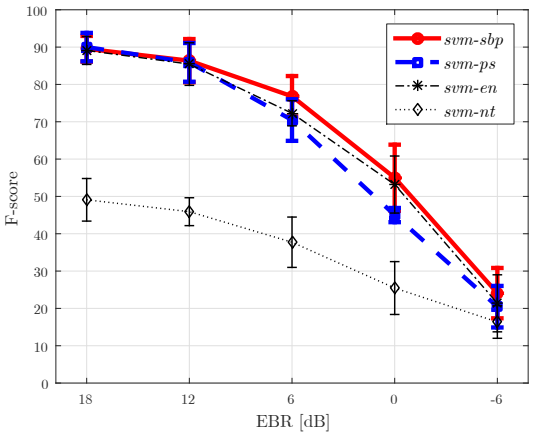



Figure 4.8: Classification performance for wrongly segmented events considering the SED-SA training set using our proposed transformation method with different T-F features and an SVM classifier.

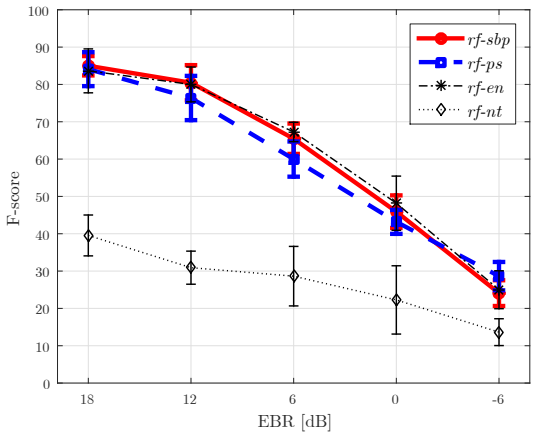



Figure 4.9: Classification performance for wrongly segmented events considering the SED-SA training set using our proposed transformation method with different T-F features and a Random Forest classifier.

previously described. Table 4.1 shows the reference performance achieved over the original noiseless and accurately isolated events, i.e. when there is a match between training and test conditions.

The curves represent the degradation of the different classification schemes varying the level of background noise. As expected, while the analyzed systems perform satisfactorily at ideal conditions (with the exception of *hmm*), the performance gets degraded as the EBR decreases. For exact segmentation, the best performing system is *snet*, showing the best robustness against noise. Note, however, that there is an abrupt drop in performance even for the highest EBR level when segmentation errors are present (Figure 4.7), and the *snet* system is outperformed by *svm-sbp*, which corresponds to the SVM classifier using our proposed transformation with *sbp* features. In fact, the $F1$ of *snet* decreases from 93% to 60% in the best scenario (18dB), demonstrating the influence of segmentation errors in the classification performance.

In order to study the performance of different feature sets used in the trace analysis stage, Figure 4.8 shows the $F1$ results using the SVM classifier under noisy and imperfect segmentation (similar to Figure 4.7). The systems that make use of our proposed approach behave robustly under non-ideal conditions, decreasing its performance as the background noise power is similar to that of the event. In that case, all the methods tend to a very low performance value, where the actual events are hardly distinguishable from the noise. Notice that, although all three methods perform similarly, the best results are obtained for *svm-sbp*.

To study the performance of our proposed trace-based transformation with a different back-end classifier, Figure 4.9 evaluates the same alternatives using Random Forests. It can be observed that very similar tendencies are obtained but the performance is slightly worse than with SVM. Given these results, the following subsections consider only the use of SVMs for classification.

Table 4.2: $F1$ performance values for the SED-SA testing set under different error segmentation scenarios.

| | Perfect (0%) | | | Isolated events (100%) | | | Continuous (50%) | | |
|---|---|---|---|---|---|---|---|---|---|
| EBR (dB) | 6 | 0 | -6 | 6 | 0 | -6 | 6 | 0 | -6 |
| *svm-sbp* | 75.7 | 50.5 | 30.3 | **67.2** (1.1) | **38.1** (1.2) | 20.7 (2.6) | **72.1** (0.7) | **46.5** (0.7) | 25.7 (1.4) |
| *svm-ps* | 75.7 | 57.6 | 34.3 | 55.7 (1.5) | 36.2 (2.5) | **23.1** (1.2) | 67.2 (1.1) | 45.8 (1.8) | 24.4 (0.8) |
| *svm-en* | 74.7 | 52.5 | 28.3 | 64.4 (1.6) | 34.7 (1.5) | 19.0 (2.0) | 70.0 (1.4) | 43.4 (1.5) | 23.0 (0.8) |
| *svm-nt* | 70.7 | 45.4 | 23.2 | 31.4 (2.6) | 16.2 (2.2) | 13.3 (0.8) | 56.1 (2.0) | 28.5 (1.1) | 14.7 (0.7) |
| *snet* | **87.9** | **80.8** | **59.6** | 39.0 (2.7) | 26.1 (1.3) | 12.6 (1.7) | 53.6 (4.9) | 43.3 (5.4) | **29.5** (3.9) |

### Performance on the test dataset

Given the above results using the SED-SA training set only, the performance over the SED-SA test set considering isolated audio events is evaluated taking into account the best performing systems, i.e. *snet* and all the SVM-based systems trained with the whole SED-SA training set. The results are shown in the three middle columns of Table 4.2, labeled as "*Isolated events*". Here segmentation errors are again generated as in the previous experiments, considering a 100% segmentation error scheme, so that the events to be classified contain portions of inactivity where only background noise is present. In this case, 10 random realizations are considered for each event in order to study the F1 variability, as shown in Figure 4.10(a). Note that, while the extracted excerpt (gray-shaded region) has always the same length, the actual location of the event varies randomly between the two extreme cases shown in the figure.

For EBR= 6dB, the best system (*svm-sbp*) outperforms in more than 35% the performance of *svm-nt* (no-trace analysis) and in more than 28% the one of *snet*. The improvement decreases for lower EBRs, where the *svm-ps* obtains the best performance. In all cases, the proposed transform alternatives outperform always *svm-nt* system, even for the lowest EBR. Note also that the results for this dataset are, in general, consistent with the trend observed for the SED-SA training set. The same results are graphically shown in Figure. 4.11. As a reference, the first three columns labeled as "*Perfect*" in Table 4.2 show the performance when the events are perfectly segmented. As expected, *snet* is the best performing system, showing a behavior similar to the one in Figure 4.6.

## 4.4.2 Robustness to continuous-stream audio event detection

In this second round of experiments, test events from the two databases have been considered. First, the scripts of the SED-SA testing set are processed as if they had been continuously acquired by the AED system. Ground-truth annotations for the onset and offset times are contaminated with noise in order to simulate the performance of an imperfect segmentation stage, obtaining wrongly segmented audio events from the continuous audio test streams. It is important here to note that only the class label is used to evaluate the classification performance, ignoring onset/offset values.

We considered a segmentation scenario where the complete events are always extracted without cuts, but which may contain parts from neighboring events happening close in time. Figure 4.10(b) shows one example indicating the signal parts that could be included by fol-

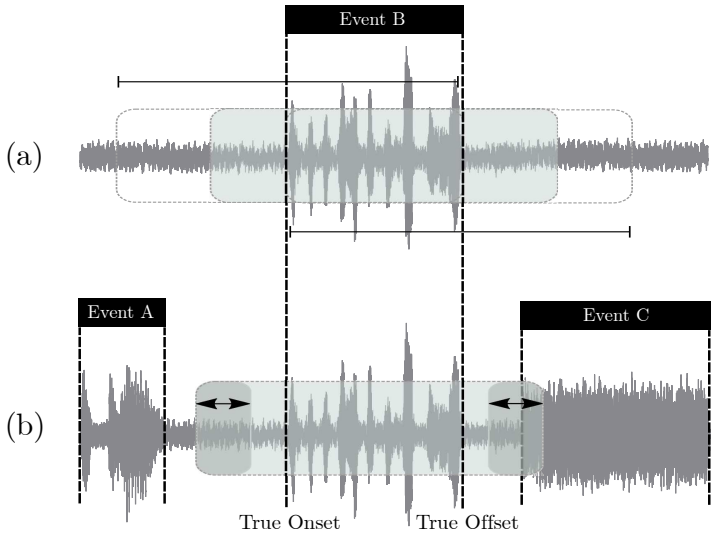



Figure 4.10: Test scenarios. (a) Isolated events, with 100% segmentation error. The gray-shaded region indicates the randomly extracted audio excerpt varying between the two shown extreme cases. (b) Continuous-stream, where the extracted excerpt may include parts from closely-spaced events under a variable segmentation error.

lowing the segmentation process when extracting Event B. Note that parts of Event C could be incorrectly extracted as part of this process.

The experiments are carried out by generating 10 different random realizations of segmentation errors for each extracted event. The contaminated onsets/offsets are uniformly varied in a range up to a quarter of the total event length which leads up to a 50% segmentation error. The three scripts are also analyzed for the three different EBR conditions already present in the test scripts.

Table 4.2 collects the results for this scenario in the columns labeled as "*Continuous*", showing the average $F1$ values together with their standard deviation in parenthesis, computed over the total number of realizations. The results show an improvement of $F1$ when using our proposed approach, with a maximum difference of 18% using *svm-sbp* at EBR= 0dB, with respect to *svm-nt*. Even though the improvement is not as high as in the experiments performed in Section 4.4.1, the performance values are actually higher. This is probably due to the amount of segmentation error considered, which is smaller in average for this case. In this scenario, the *snet* system achieves the best results for EBR= $-6$dB, however, it is important to notice that its variability is considerably higher than the rest. The results for this scenario are also graphically shown in Figure 4.12.

The same experimentation has been carried out using the more realistic TUT Rare Sound Event dataset from DCASE17. The main difference is that each event has been classified only once so we have only one performance measure. On the other hand we consider three different levels of segmentation error apart from perfectly segmented events: 40%, 60% and 100%. The corresponding results are shown in Table 4.3 where very similar trends to the ones above can be observed. It is worth noting that *snet* is the overall best option for the lowest EBR level. Nonetheless, the proposed *svm-sbp* method exhibits in general a better behavior as degradation increases.

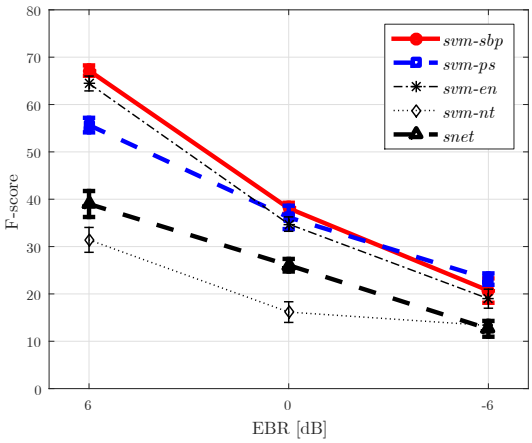



Figure 4.11: Classification performance for wrongly segmented events considering the SED-SA testing set. The events are isolated as in Figure 4.10(a).

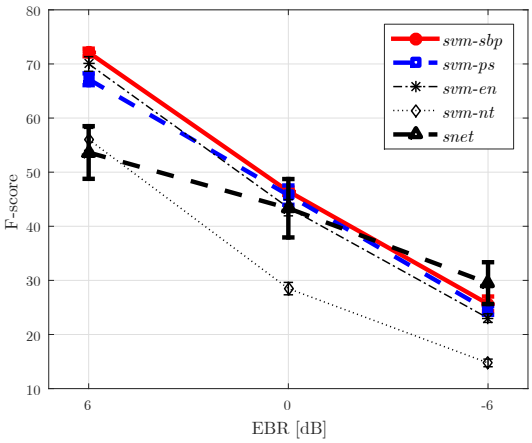



Figure 4.12: Classification performance for wrongly segmented events considering the SED-SA testing set. The events are extracted from continuous audio streams as in Figure4.10(b).

Table 4.3: $F1$ performance values for the TUT Rare Sound Events under different amounts of segmentation error.

| | Perfect (0%) | | | Continuous (40%) | | | Continuous (60%) | | | Continuous (100%) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EBR (dB) | 6 | 0 | -6 | 6 | 0 | -6 | 6 | 0 | -6 | 6 | 0 | -6 |
| *svm-sbp* | 85.46 | 83.15 | 66.27 | 83.70 | **86.14** | **67.46** | 85.02 | **84.64** | 63.89 | 85.90 | **83.52** | **64.68** |
| *svm-nt* | 87.67 | 79.40 | 56.35 | 80.18 | 65.54 | 50.79 | 83.70 | 71.16 | 51.59 | 83.26 | 65.54 | 50.00 |
| *snet* | **89.43** | **87.27** | **73.02** | **88.55** | 74.53 | 57.54 | **89.87** | 80.90 | **66.67** | **89.87** | 77.15 | 60.32 |

As observed in both tables, in general, our proposed approach achieves always better performance than the *svm-nt* conventional scheme. Deciding which is the best set of features for trace analysis is not easy, since their performance seems to be dependent on the specific test case and sound classes. For most cases the *svm-sbp* seems to be more consistent in its behavior over different noisy scenarios and segmentation errors.

### 4.4.3 Discussion

The analysis of the previous results confirm the robustness of our proposed event representation method both to segmentation errors and noise. It is interesting to emphasize that, while the proposed method achieves a considerable performance gain, small differences are observed among the different feature sets used for trace analysis. Although the *svm-en* feature provides always the worst performance, the selection of a given feature set may depend on the computational constraints of the application at hand. For example, it is observed that satisfactory results are already obtained by *svm-en* with no need for computing any frequency coefficients. As a result, this representation may be convenient in applications where nodes with limited energy resources are to be employed [28].

Different alternative event representations for SER based on uniform distance subsampling guided by a reduced feature space, have been investigated. The proposed method

makes use of a non-linear transformation that maps the event from a uniform time sampling scheme to a uniform distance sampling considering different T-F features. The trace analysis stage is included in order to define the mapping from one representation to the other.

We compared the use of the proposed approach with the classical short-term temporal framing performed in Chapter 3, where the statistics are computed over a fixed-length sliding window. Different feature sets to embed the temporal evolution of the event have been analyzed, evaluating their classification accuracy when background noise and segmentation errors are present. The results have shown that the proposed event representation performs consistently better than the classical approach for a wide range of EBR values and different types of segmentation errors. In this context, a performance gain is achieved when extracting incomplete events or when inaccurate onset/offset time estimates lead to excessive background noise at the beginning or end of the extracted excerpts.

The use of the considered SVM framework is justified due to the small size of the database as compared to other machine learning works where a large amount of training/test data is available. Deep models have shown great generalization capabilities, when trained with large amount of training data. However, deep features are considerably affected when adverse acoustic conditions are considered, similarly to what happens to the hand-crafted features. The proposed adaptive fixed-length representation has shown to be robust against different acoustic conditions, motivating the implementation of our approach in combination with the generalization capability of deep features.

## 4.5 Conclusion

In this chapter, we investigated an alternative event representation for acoustic event classification based on uniform distance sampling over a reduced feature space. The method makes use of a non-linear transformation that maps the event from a uniform time sampling scheme to a uniform distance sampling considering different T-F features. A trace analysis stage is included in order to define the mapping from one representation to the other. We compared the use of the proposed approach with the classical short-term temporal framing, where the statistics are obtained over a fixed-length sliding window. Different feature sets to embed the temporal evolution of the event have been analyzed, evaluating their classification accuracy when background noise and segmentation errors are present. The results have shown that the proposed event representation performs consistently better than the classical approach for a wide range of EBR values and different types of segmentation errors. In this context, a performance gain is achieved when extracting incomplete events or when inaccurate onset/offset time estimates lead to excessive background noise at the beginning or end of the extracted excerpts.

An obvious extension for the T-F space used in trace analysis could be to learn an optimal subspace from data. Note, however, that this would require a significant amount of data which is not available in our present context. In fact, it should be taken into account that the considered SVM framework is justified due to the small size of the database as compared to other machine learning works where a huge amount of training/test data is available.

# Chapter 5

# Distance-Based Adaptive Pooling in Convolutional Nets

The effectiveness of an acoustic recognition model is often hampered by the limited access to a large enough strongly labeled database, providing insufficient coverage of a representative sample of the variability present in the acoustic data. Classification models trained using small, biased or insufficiently varied databases can be overfitted and unable to generalize well to unseen examples. There are several techniques to mitigate this problem, such as data augmentation methods. However, by increasing the training data the learning process becomes also more complex and time consuming.

During this thesis, it has been studied how feature selection affects the performance of the classifier in different ways. Accuracy is severely affected by the training/test condition mismatch and, whether hand-crafted features or deep features are used, improving the robustness to such mismatch is important to develop SER systems working in realistic environments. Although deep features avoid the use of complex feature selection techniques, they do not present good generalization capabilities under mismatched conditions.

In the previous chapter, the effect of segmentation errors was mitigated by applying an activity-based non-linear transformation to a selected set of hand-crafted features using the concept of trace. In this new chapter, we present a pooling layer aimed at compensating non-relevant information of audio events by applying an adaptive transformation of the convolutional feature maps in the temporal axis by following a similar rationale. To this end, we propose the use of the trace extracted from internal layer activations to include activity information within the learning process of CNNs. Specifically, the activations from internal layers are conveniently downsampled following an adaptive non-linear transformation, guided by a uniform segmentation of the trace at a given network depth. The experiments conducted during this work over different datasets, have shown significant performance improvements when the proposed layer is added to a baseline model, resulting in systems that generalize better to segmentation errors and noise.

## 5.1 Temporal summarization and weakly segmented data

One of the challenges of working with audio data is the variable length nature of real-life sounds. Although a stream of audio data can be analyzed by a CNN in a sliding window

fashion, the network is designed to have a fixed-length input, assuming that relevant features can be located anywhere within an input segment. As a result, the performance may strongly depend on how specific audio segments are extracted from a training dataset and how the test data windowing resembles that of the training data. In this context, different approaches are widely used to deal with the length variability of intra- and inter-class sounds. For example, short audio files are zero-padded or replicated up to a given length, while longer files are trimmed [43], being the fixed length a trade-off based on the average duration of the available audio samples. Another approach is to perform a frame-by-frame classification by means of a sliding-window [75].

### 5.1.1 Temporal aggregation schemes

Even if the architecture of the model is adapted to the properties of the training data, the testing examples most probably will have different characteristics. An alternative to the requirement of having a fixed-length input, due to the use of fully connected layers, is the use of global pooling strategies. These layers aggregate the temporal information extracted from the previous layers by mapping regions to a single value. Figure 5.1 shows a comparison between these two layers. The advantage of using the global pooling layer is that it accepts variable length input while being invariant to small displacements.

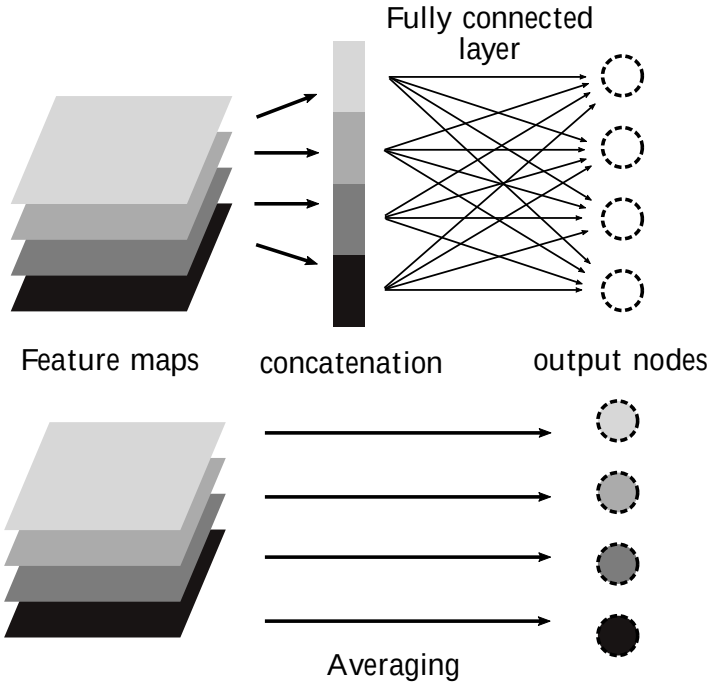



Figure 5.1: Comparison between fully connected and global average pooling layers.

The pooling layer was introduced to make the network robust against small translations. If the representation is invariant to small amount of displacements at the input, the values of the pooled outputs will not change. This was inspired by face recognition task, where it was important to detect if in the image there was a pair of eyes, independently of the location of those eyes. The choice of extracting the mean statistics is because of how the pooling operation propagates the gradient information. While max pooling assigns all the responsibility to the largest instance, averaging gives equal responsibility to all the instances. Therefore, in contrast to max pooling, all the instances contribute to update the parameters of the network. Global average pooling (GAP) was proposed in [87] to avoid the use

of fully connected layers as a final output given their tendency to overfit. However, this layer is not well suited for modeling sequential data. Recurrent neural networks (RNN) were introduced to use the activation from earlier frames in order to model sequential data such as speech [52] or handwriting recognition [53]. By adding a feedback loop, the network is able to learn from sequences, capturing temporal information as introduced in Section 2.7. Historically, these networks have been difficult to train, but the improvements in hardware capabilities to train such models have popularized their use for solving detection-related task.

### 5.1.2 Weakly segmented data

Nonetheless, the main drawback of the techniques described above is usually on the mismatch between the training and test data: while training samples are usually perfectly isolated due to a supervised annotation process, test data is usually obtained in real-world conditions, containing variable-length segments, weakly segmented events and background noise. Recently, bigger human-labeled datasets have been made available for the audio research community, such as AudioSet [46], a large-scale database with more than 600 classes. However, training samples are annotated using weak labels, i.e. only the presence or absence of the events are indicated, with no information about their onset and offset times. This can be very challenging, since the audio files are 10 seconds long whereas the events of interest may be very short in comparison to the total excerpt duration. Moreover, marking the timestamps of the events can be very ambiguous given the uncertainty present in humans annotators, which makes the annotation process much harder. Additionally, although multimedia content is widely available through the Internet and it is relatively easy to obtain and weakly annotate it using its metadata information, the process unavoidably introduces noise to the data. For this reason, it is interesting to design CNNs robust against length-related mismatches between training and testing data and against errors in onset/offset labeling.

The problem of robust classification of weakly segmented audio events with fixed-length classifiers has been already addressed in Chapter 4. The proposed method followed a trace segmentation strategy in order to embed the temporal evolution of audio events into fixed-length feature vectors used for classification. A key point of such method was the use of two different feature spaces: one for trace analysis which presumably captures the temporal behavior of the event, and the actual feature space enhancing class discrimination. The so-called trace refers here to the accumulated frame-to-frame Euclidean distance in the analysis feature space, which can be roughly understood as a measure of temporal activity within such space. The computed trace is then used as a reference for downsampling variable-length feature vectors to a fixed-length representation suitable for classification.

Following a similar principle, the approach presented in this chapter, makes use of the trace extracted from internal layer activations to include activity information within the learning process of convolutional neural networks. Specifically, the activations from internal layers are conveniently downsampled following an adaptive non-linear transformation, guided by a uniform segmentation of the trace at a given network depth.

## 5.2 Proposed approach

This section describes the proposed distance-based pooling strategy based on trace analysis. As it will be next presented, the approach can be easily incorporated into CNNs to perform an adaptive downsampling procedure across the network based on the temporal activity of

the event.

### 5.2.1 Layer input, output and notations

Let $\mathbf{x}^l \in \mathbb{R}^{H^l \times W^l \times D^l}$ be the input to the $l$-th layer, where $H^l$ denotes the height, $W^l$ the width and $D^l$ the number of channels of the input. In contrast to image processing networks, where color inputs make use of 3 channels, an end-to-end audio-oriented network has usually a single channel ($D^l = 1$). In general, $\mathbf{x}^l = \mathbf{y}^{l-1}$, i.e. the input of the $l$-th layer corresponds to the output of the previous one, which is assumed here to be a convolutional layer (Conv-1D). Therefore, the $H^l$ dimension is related to the temporal length of the convolutional output given a selected kernel size and stride, while $W^l$ denotes the number of filters of the layer. Note that the columns of $\mathbf{x}^l$ correspond to the different feature maps extracted from the learned filter outputs at the layer $l-1$.

Pooling layers operate upon $\mathbf{x}^l$ and they do not require parameter learning. They operate over the $H^l \times W^l$ elements of the input by mapping subregions into a single number, performing some kind of signal downsampling. Conventional pooling schemes are max pooling and average pooling. In max pooling, the pooling operator maps a subregion to its maximum value, while the average pooling maps a subregion to its average value. The following subsection presents the proposed distance-based adaptive pooling. In contrast to conventional schemes where subregions remain fix given a defined kernel size and stride, the proposed method considers adaptive subregions that are redefined from $\mathbf{x}^l$ for each training and test example with the aim of accommodating the temporal activity of the event.

### 5.2.2 Distance-based pooling

Consider the input to the $l$-th layer, $\mathbf{x} \in \mathbb{R}^{H \times W}$, where the superindex $l$ and the channel dimension have been omitted for the sake of notation clarity. Expressed in terms of its columns, we can write $\mathbf{x}$ as

$$\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_W], \tag{5.1}$$

where $\mathbf{x}_j \in \mathbb{R}^{H \times 1}$, $j = 1, \ldots, W$, are the feature (1D) maps resulting from the set of filters of the previous layer. Similarly, we can interpret such input as a collection of row vectors as follows:

$$\mathbf{x} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_H \end{bmatrix}, \tag{5.2}$$

where $\mathbf{f}_i \in \mathbb{R}^{1 \times W}$, $i = 1, \ldots, H$, can be considered as feature vectors extracted at the $l$-th layer of the network within the $i$-th temporal window:

$$\mathbf{f}_i = \left[ f_1^{(i)}, f_2^{(i)}, \ldots, f_W^{(i)} \right]. \tag{5.3}$$

Note that the individual features extracted from the $i$-th temporal frame are denoted as $f_j^{(i)}$, $j = 1, \ldots, W$, where $W$ is the number of filters of the previous layer.

#### Trace

The temporal evolution of an audio event can be assumed to create a trajectory in the $W$-dimensional feature space, which starts at the spatial coordinates $\mathbf{f}_1$, moves to $\mathbf{f}_2$ and continues until reaching $\mathbf{f}_H$. This temporal evolution can be summarized into the so-called

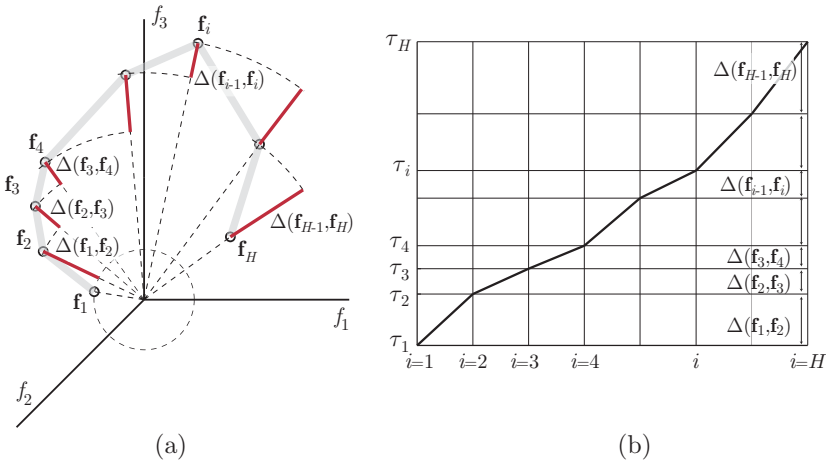



Figure 5.2: Generation of the trace. (a) Temporal trajectory of activations in the feature space. (b) Trace resulting from magnitude differences.

trace of the event, which represents the accumulated Euclidean distances between consecutive temporal frames. In [95], different sets of power-related features were proposed to translate the temporal evolution of audio events into a trace. In fact, the results demonstrated that the trace computed from the short-time energy of the event was a sufficient activity indicator. Given the relatively large amounts of data needed to train deep neural networks, we propose here a simplified trace aimed at alleviating computations and accelerating the training of the network. Thus, instead of using an alternative set of features to compute the trace, it is computed directly from the magnitude of layer activations.

Taking into account the above considerations, the trace of the event, $\boldsymbol{\tau} \in \mathbb{R}^H$, is given by

$$\boldsymbol{\tau} = [\tau_1, \tau_2, \ldots, \tau_H]^T, \tag{5.4}$$

where the elements $\tau_i \in \mathbb{R}$ of the vector are defined as

$$\tau_i = \begin{cases} 0, & i = 1. \\ \sum_{t=2}^{i} |\Delta(\mathbf{f}_{t-1}, \mathbf{f}_t)|, & i = 2, \ldots, H, \end{cases} \tag{5.5}$$

where

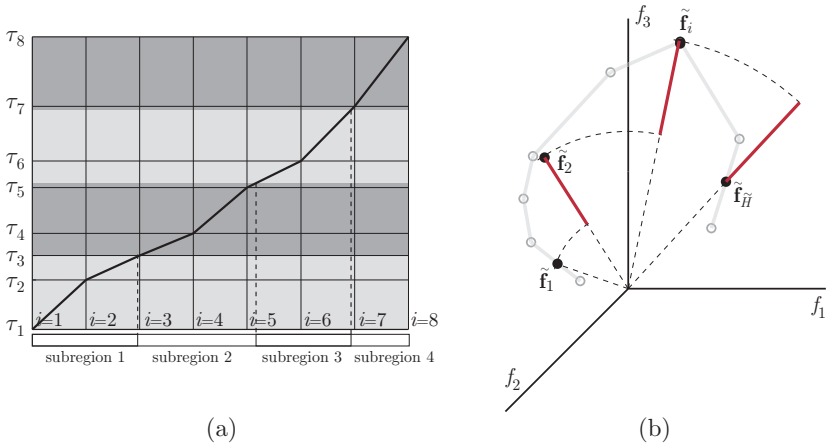
$$\Delta(\mathbf{f}_{i-1}, \mathbf{f}_i) \triangleq \|\mathbf{f}_i\| - \|\mathbf{f}_{i-1}\| \tag{5.6}$$

represents the magnitude difference between two consecutive frames.

The process is exemplified in Figure 5.2 for a simple three-dimensional ($W = 3$) case. In (a), the temporal trajectory of feature vectors $\mathbf{f}_i$ extracted from layer activations is represented in gray. The magnitude differences corresponding to consecutive frames are depicted in red. The resulting trace vector $\boldsymbol{\tau}$ is shown in (b), where the elements of $\tau_i$ are obtained from such accumulated differences.

### Downsampling

The objective now is to obtain a resampled version of the layer input $\mathbf{x}$ using the trace information $\boldsymbol{\tau}$, resulting in the output $\mathbf{y} \in \mathbb{R}^{\tilde{H} \times W}$. The rationale underlying trace-based downsampling resides in the fact that the information contained in $\mathbf{x}$ comes from a uniform temporal sampling of the event, since its rows $\mathbf{f}_i$ are extracted from analyzing the input audio




(a) (b)

Figure 5.3: Trace-based pooling. (a) Uniform segments within the trace (gray regions) define the averaging subregions. (b) Output resulting from averaging the points pertaining to each subregion. The magnitude differences are now balanced.

sequence in a sliding window fashion. However, the proposed distance-based pooling intends to downsample the input with equidistant spatial segments along its trace.

The output is calculated by dividing the total spatial length of the trace, $\tau_H$, into $\tilde{H}$ uniformly spaced segments, where the length of each spatial segment is:

$$L = \frac{\tau_H}{\tilde{H}}. \tag{5.7}$$

The above length is then used to group the original feature vectors $\mathbf{f}_i$ into $\tilde{H}$ sets corresponding to different subregions:

$$\mathcal{R}_k = \{\mathbf{f}_i : (k-1)L \leq \tau_i \leq kL\} \quad k = 1, \ldots, \tilde{H}. \tag{5.8}$$

The output feature vectors are obtained as the average of the original feature vectors contained in each subregion, i.e.

$$\tilde{\mathbf{f}}_k = \frac{1}{|\mathcal{R}_k|} \sum_{\mathbf{f}_i \in \mathcal{R}_k} \mathbf{f}_i \quad k = 1, \ldots, \tilde{H}, \tag{5.9}$$

where $|\mathcal{R}_k|$ denotes the cardinality of the set $\mathcal{R}_k$. The output of the proposed pooling layer is therefore given by

$$\mathbf{y} = \begin{bmatrix} \tilde{\mathbf{f}}_1 \\ \tilde{\mathbf{f}}_2 \\ \vdots \\ \tilde{\mathbf{f}}_{\tilde{H}} \end{bmatrix}. \tag{5.10}$$

Following the example of Figure 5.2, the downsampling process by using the computed trace is shown in Figure 5.3. In the example, the number of feature vectors has been reduced by half using $\tilde{H} = 4$. In (a), the uniform sections along the trace are highlighted in gray, resulting in 4 non-uniform subregions comprising a different number of feature vectors. For example, while subregion 2 comprises three feature vectors ($i = 3, 4, 5$), the third subregion only contains one feature vector ($i = 6$). The result of the pooling operation is shown in (b), where the new feature vectors are marked as black dots in the feature space. Note that the new vectors, follow a trajectory similar to the original one, but now the magnitude differences (marked in red) are approximately equal.

### Example

To visualize better how the proposed pooling affects the information propagated through the network, Figure 5.4 shows a real example from the baseline architecture used in this work, which is described in detail in Section 5.3. The original input sequence containing the event is shown in (a), where it can be appreciated from the envelope that the event onset happens around 2.26 seconds. The activations resulting from an internal layer of the network are shown in (b). The onset within such internal representation appears approximately at frame 20 (marked with a dashed rectangle). The (normalized) trace computed from (b) is in (c), where the different onsets making up the event appear as slope changes through the trace. The result of applying conventional average pooling to (b) is shown in (d), while the result of applying the proposed trace-based pooling is represented in (e). While both pooling schemes reduce the length of the input from $H = 77$ to $\tilde{H} = 19$, it can be clearly observed how the adaptive pooling layer has focused its attention on those activations where the trace of the event increases more rapidly.

### Backward propagation

Although pooling layers do not need to learn any parameters, error backward propagation has to be considered during the training process. As it happens with conventional max pooling or average pooling, the forward propagation step results in a pooling subregion being reduced to a single value. In conventional max pooling, this value is the one corresponding to the "winning unit", and to keep appropriate track of the error acquired by such unit, its index is stored during the forward pass and used for gradient routing during the backward pass.

In our proposed pooling scheme, the error should be propagated back by sharing among all the units coming from the different pooling blocks or subregions. Since subregions change for each training example, it is needed to store during the forward pass the elements of each set $\mathcal{R}_k$. Thus, the error from the $k$-th set must be multiplied by $\frac{1}{|\mathcal{R}_k|}$, assigning the resulting value to all the units making out the set.

## 5.3 Experiments

This section describes the experimental methodology followed to assess the performance of the proposed pooling scheme, including the baseline model, the public datasets and the performance metrics used to confirm the validity of the presented approach.

### 5.3.1 The SoundNet baseline system

The baseline end-to-end classification model considered in this work is based on the SoundNet [12] CNN architecture, which uses max pooling and ReLU activations after each convolutional layer. The original SoundNet model consists of eight convolutional layers, as has been described in Section 2.7. However, for this study, we selected up to the seventh layer, as depicted in Figure 5.5. The layer parameters are specified in Table 5.1.

An advantage of the above baseline model is the possibility to feed input audio sequences of varying length. The network was originally trained with variable-length videos and, given the importance of the temporal evolution of sound events in our proposed approach, such architecture is considered to be very well aligned to our problem.

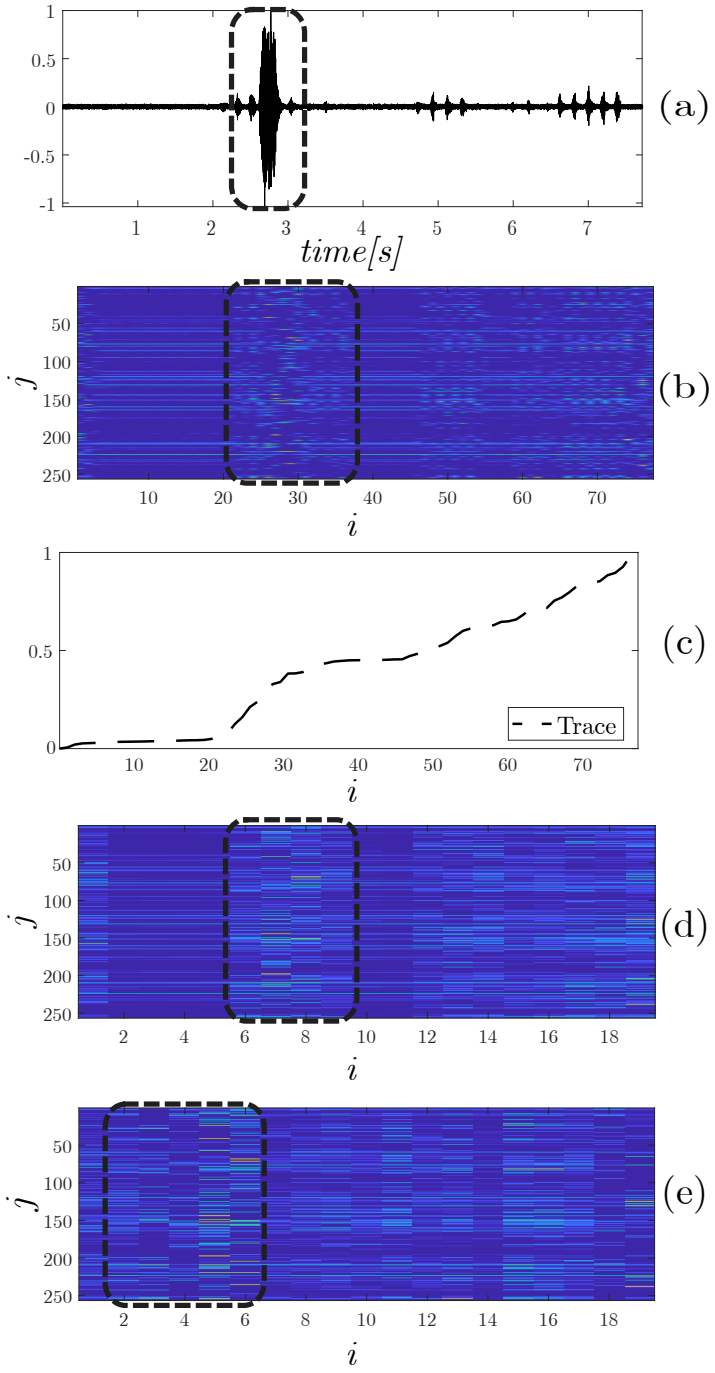



Figure 5.4: Example of the proposed pooling scheme applied over an audio event. (a) Original audio event. (b) Activations from *conv*5 layer (dimensions $W = 256$, $H = 77$). (c) Trace of the event computed from the activations ($\tilde{H} = 19$). (d) Result from applying conventional average pooling. (e) Result from applying the proposed distance-based pooling using the trace. Note that the layer outputs are transposed to identify better the temporal axis and its correspondence to the original audio envelope.

Table 5.1: SoundNet layer parameters.

| Layer | conv1 | pool1 | conv2 | pool2 | conv3 | conv4 | conv5 | pool5 | conv6 | conv7 |
|---|---|---|---|---|---|---|---|---|---|---|
| $W$ | 16 | (16) | 32 | (32) | 64 | 128 | 256 | (256) | 512 | 1024 |
| kernel size | 64 | 8 | 32 | 8 | 16 | 8 | 4 | 4 | 4 | 4 |
| stride | 2 | 8 | 2 | 8 | 2 | 2 | 2 | 4 | 2 | 2 |
| $-\log_2(H)$ | 1 | 4 | 5 | 8 | 9 | 10 | 11 | 13 | 14 | 15 |

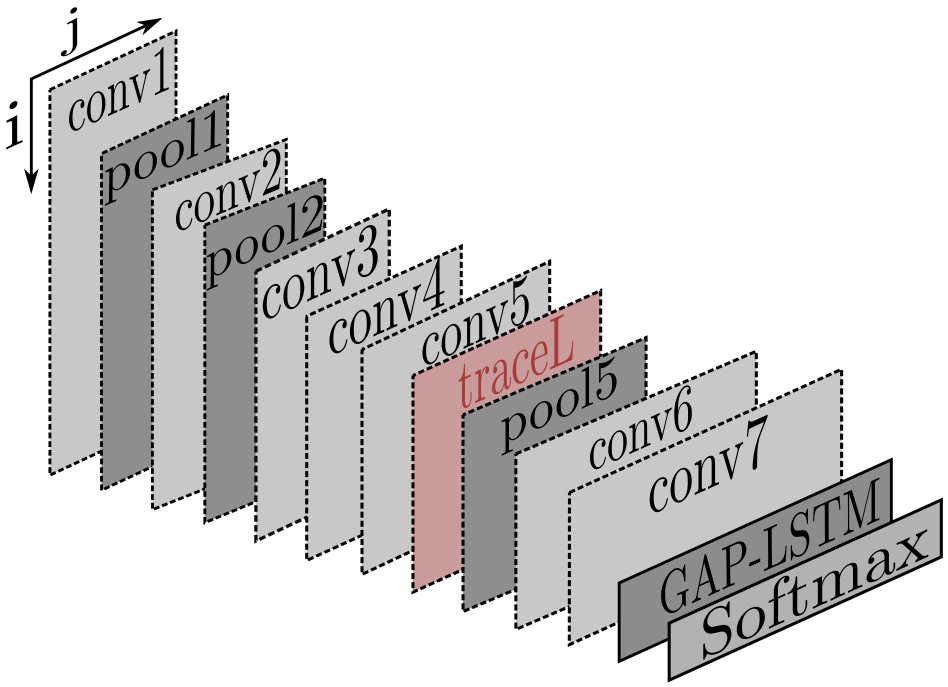



Figure 5.5: Architecture of our model based on the SoundNet network.

### Incorporation of the trace-based adaptive pooling layer

Selecting the best layer for incorporating our proposed approach is not straightforward. The first layers of the network contain more temporal information but they may also encode more general properties of the input signal. In contrast, the last layers contain more specific information but the temporal granularity is really scarce. Several experiments have been conducted in order to determine the most appropriate depth for inserting our distance-based approach, concluding that intermediate layers provide the best results. For the specific SoundNet structure, we selected the best performing depth, which is given by the fifth convolutional layer (marked in red in Figure 5.5).

### Classification

The assignment of class labels to input audio sequences is performed by means of two widely used approaches: global average pooling (GAP) [87] or LSTM [59], both followed by a feed-forward layer using softmax activation. The use of global average pooling allows to aggregate the final activations in the temporal axis in a simple way without the need to add extra parameters to the network. Alternatively, feeding CNN outputs to recurrent layers has been

shown to be a powerful approach for sound event detection [169]. Thus, we also adopt for our experiments a standard LSTM cell with 32 or 64 units before the final feedforward layer.

#### Training

The training is performed by feeding the raw audio files directly to the network, using a batch size of 32. The optimization function used during training is categorical cross-entropy, and a maximum of 200 epochs is necessary with early stopping when the loss does not decrease during 20 consecutive epochs. The system is evaluated obtaining the *Accuracy* metric, which is obtained as the number of correctly predicted labels divided by the total number of predictions. In all the experiments, the Adam optimizer [67] was used, with a learning rate of 0.0001 for **ESC** and 0.00001 for **URBAN**.

### 5.3.2 Datasets

The proposed system is evaluated over two well-known audio datasets, ESC-50 (**ESC**) and UrbanSound8K (**URBAN**) with the aim of facilitating the comparison of the presented results with the ones of other state-of-the-art systems. Moreover, the selected datasets are also representative in terms of size, since one contains a small amount of data with a considerable number of classes and the other has a small number of classes but a large amount of samples per class. Both datasets have been previously defined in Section 5.3.2. The audios from those datasets are previously resampled to 22050 to match the original SoundNet system.

### 5.3.3 Generation of adverse conditions

The described databases are comprised by isolated sound events. In general, training datasets are carefully built and annotated, but when the system has to perform in realistic scenarios, the test examples are not usually properly segmented and isolated from background noises. Thus, to evaluate the proposed method, we modify the datasets at test time to simulate different segmentation errors and varying noise conditions. This is done by adding different levels of additive background noise of the Brownian type, as in [126]. The different EBR levels and the segmentation errors are simulated as defined in Section 3.3.1. The EBR levels considered are, EBR $\in \{18, 12, 6, 0\}$ dB. For the segmentation errors, we assume weakly segmented events by artificially enlarging their original duration with noise. The amount of pre and post segmentation noise is varied randomly to add a total of 2, 3, 4 or 5 seconds of background noise.

## 5.4 Results and discussion

This section discusses the results obtained for the experiments described in Section 5.3. Given the elevate number of conditions changing over the experiments, the analysis is divided in three parts. The first part provides reference performance values for the case when the test conditions match those of the training ones, i.e. when the test examples are directly used from the selected public databases without any additional degradation. The second part analyzes the performance of the method for varying degrees of segmentation errors. Finally, the third set of experiments analyzes the accuracy of the system for a range of background noise levels considering a fixed percentage of relative segmentation error.

Table 5.2: Reference performance for matching conditions.

| | | **C7_gap** | **C6_gap** | **C7_64** | **C6_64** | **C7_32** | **C6_32** |
|---|---|---|---|---|---|---|---|
| *ESC* | NT | 72,54 (4,45) | 71,37 (4,70) | 70,94 (3,19) | 68,42 (4,52) | 68,42(4,01) | 65,13 (5,30) |
| | T | **72,58** (3,01) | 71,32 (3,18) | 71,85 (3,05) | 69,30 (3,74) | 71,68 (2,58) | 66,79 (3,18) |
| *URBAN* | NT | 72,54 (6,27) | 71,95 (6,17) | 70,66 (6,33) | **74,48** (3,91) | 72,06 (6,06) | 74,14 (4,37) |
| | T | 72,23 (6,25) | 69,29 (6,96) | 71,43 (4,87) | 73,65 (4,44) | 72,13 (4,73) | 73,77 (5,07) |

Different network configurations are tested to provide results obtained by different network depths and temporal aggregation schemes. Thus, the nomenclature used throughout all the experiments is as follows:

- Global Average Pooling (GAP), selecting layers up to *conv*6 and *conv*7: {C6_gap, C7_gap}.
- LSTM, varying the number of neurons between 32 and 64 and selecting layers up to *conv*6 and *conv*7: {C6_32, C6_64, C7_32, C7_64}.

For all the above systems, results are presented for the case where our proposed adaptive pooling layer is included (T) and when it is not (NT).

### 5.4.1 Reference Performance with Matching Conditions

Table 5.2 contains the cross-validation mean accuracy values together with their corresponding standard-deviations in parentheses (5 folds for ESC and 10 folds for URBAN) when no modifications to the original datasets are included. The performance achieved is comparable to that of other state-of-the-art systems evaluated with the same datasets [115, 128]. In general, for the ESC dataset, the results tend to be slightly better for GAP than for LSTM, especially for *conv*7. For this dataset, the performance using our proposed layer (T) is also slightly better than without it (NT), providing as well some improvement in terms of variability. For the URBAN dataset, the best performances are obtained for LSTM at *conv*6. The use of our trace-based layer does not produce any remarkable improvement, but this also means that the introduction of our pooling layer does not affect negatively the performance in ideal scenarios where the test conditions resemble those of the training set.

### 5.4.2 Robustness to weak segmentation

This subsection compares the performance of the system with and without including our proposed distance-based pooling under weak segmentation conditions. To this end, the test data is modified by adding Brownian noise at EBR = 18 dB and extended in duration with different temporal lengths (2, 3, 4 and 5 seconds).

Figure 5.6 shows the results for the ESC database considering depths up to *conv*6 and *conv*7 for GAP and LSTM. Given the results of Table 5.2, only the systems with 64 units were considered for having better performance. The white bars indicate systems using our proposed approach (T), while gray bars correspond to the baseline case without trace pooling. Note that the performance gets degraded in all cases with respect to the ideal results of Table 5.2, but the systems using our proposed pooling scheme always result in better performance. This is especially true for *conv*6, where the advantages of the proposed layer become more relevant as the length of the input sequence containing the event gets smoothly

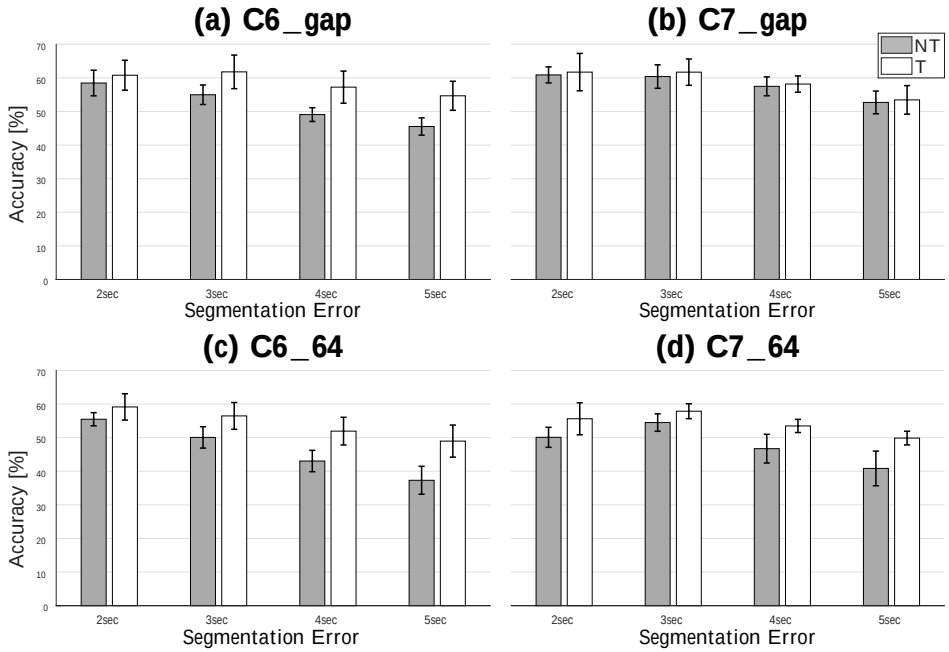



Figure 5.6: Accuracy values for the ESC dataset under weak segmentation and EBR = 18 dB.

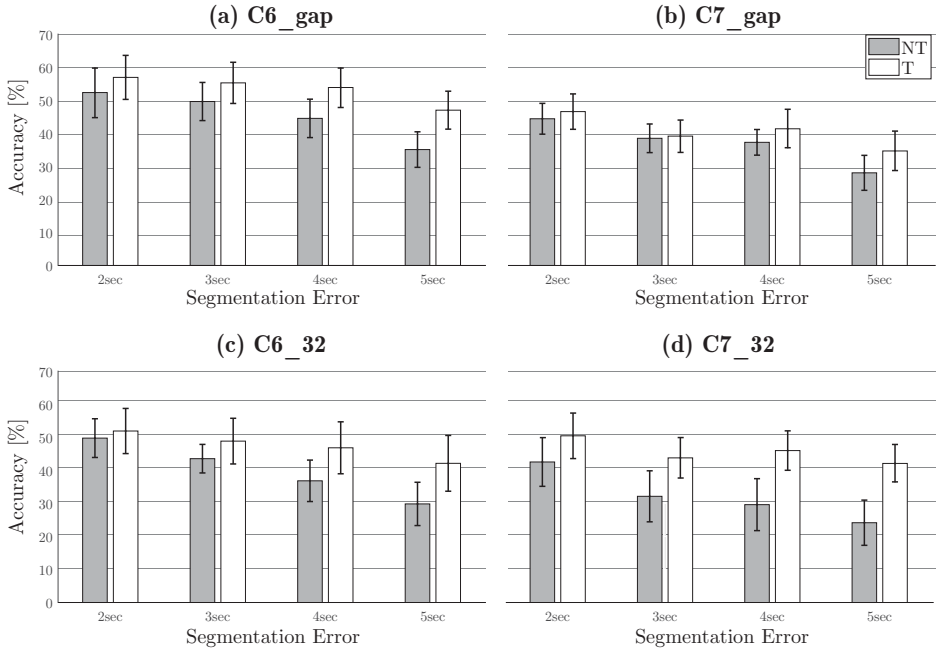



Figure 5.7: Accuracy values for the URBAN dataset under weak segmentation and EBR = 18 dB.

Table 5.3: Accuracy for ESC for different levels of EBR, brownian noise.

| | **EBR [dB]** | | | |
|---|---|---|---|---|
| **Layer** | **18** | **12** | **6** | **0** |
| C6_gap_NT | 64,48 (3,79) | 60,67 (3,88) | 55,37 (2,35) | 46,64 (1,87) |
| C7_gap_NT | 65,49 (5,66) | 63,04 (4,74) | 58,11 (3,81) | 50,72 (1,45) |
| C6_gap_T | **67,60** (4,26) | **66,98** (3,93) | 61,17 (4,77) | 49,95 (2,51) |
| C7_gap_T | 65,93 (4,32) | 65,03 (3,78) | **61,21** (3,83) | **54,16** (2,46) |

Table 5.4: Accuracy for URBAN for different levels of EBR, brownian noise.

| | **EBR [dB]** | | | |
|---|---|---|---|---|
| **Layer** | **18** | **12** | **6** | **0** |
| C6_gap_NT | 66,10 (6,91) | 65,65 (6,64) | 60,59 (6,25) | 50,71 (6,90) |
| C6_32_NT | 59,12 (5,12) | 57,28 (6,16) | 54,07 (4,94) | 48,45 (3,25) |
| C6_gap_T | **66,44** (6,16) | **66,62** (6,55) | **64,07** (6,03) | **54,75** (6,20) |
| C6_32_T | 60,49 (5,84) | 61,64 (6,20) | 58,55 (4,47) | 52,74 (5,17) |

increased. For a case of an extended duration of 5 seconds, the performance gain is higher than 10%.

The corresponding results for the URBAN dataset are shown in Figure 5.7. In this case, only the LSTM systems made up of 32 units are considered for being the best performing ones. The advantages brought by the proposed system are more relevant for this dataset, where the differences between T and NT are close to 20%in the most aggressive case (5 seconds in C7_32). As with the ESC dataset, the differences between including our pooling scheme and not including it tend to be higher as the length mismatch increases. In any case, and for all layers and temporal aggregation schemes, the T-systems outperform the NT ones.

### 5.4.3 Robustness to background noise

The robustness to different EBR conditions considering a fixed relative segmentation error of 80% is here discussed. Tables 5.3 and 5.4 show, respectively, the results for the ESC and the URBAN datasets. In both cases, only the best performing systems from Figures 5.6 (ESC) and 5.7 (URBAN) are considered. The performance differences between T (proposed) and NT systems are approximately 5% for very noisy conditions (6 dB and 0 dB), although results vary slightly depending on the network depth. Nonetheless, including the proposed pooling scheme results always in better results than the corresponding baseline system, especially when adverse conditions get stronger.

## 5.5 Conclusion

In this chapter we have proposed a distance-based pooling layer aimed at improving the performance of CNN-based models for audio event classfication in adverse scenarios. Specif-

ically, the proposal addresses the problem caused by the usual mismatch existent between training and test conditions in realistic scenarios, where weakly segmented audio events and different levels of background noise are present at test time. The proposed layer performs a non-linear transformation of the input on the temporal axis using the trace of the event, which is built considering the norm differences of layer activations across time. The process results in a uniform distance subsampling in an internal feature space that allows to propagate better the information of the actual event through the network. Results considering a widely known baseline architecture (SoundNet) and two public datasets confirm the advantages of the approach under adverse scenarios with severe training and test mismatches.

# Chapter 6

# Regression-Based Soft Event Detection

The representation of the temporal evolution from a sound event using its energy envelope, has been shown to be useful for capturing representative parts of the signal, which has inspired the use of this information to overcome the inherent subjectivity when annotating sound events. Even though the number of acoustic databases has increased, it is still very expensive to properly annotate the precise onset and offset times for each event, which leads to the so-called strongly labeled datasets. Apart of being time consuming and expensive, strong labeling has to deal with ambiguities and interpretation issues due to human annotators. To reduce the effort in the annotation process, the use of weakly labeled data for SED-related tasks has been introduced [74].

In the case of weak labeling, only information about event presence or absence of the event is considered without the necessity to locate it within an audio excerpt. This makes it considerably easier to collect new annotated data by simply extracting information from the metadata associated to the audio recordings. Following this approach, a large-scale database has been released by Google, with 632 audio classes [46]. However, the use of weakly labeled databases does not provide the level of accuracy that a system trained with strong labels can achieve. As a further alternative, synthetic mixtures can be created in order to automatically obtain the ground-truth of the events within the mixture. By training with synthetic mixtures, the uncertainty can be controlled, guaranteeing perfect and complete annotations.

The most common representation of sound events in current systems is in the form of binary activity indicators for individual sound instances. However, this is a very rough approximation of the natural activity patterns of sounds in real life. Sounds have often different non-binary activity patterns. For example, moving sources such as a car passing by or a vehicle siren, exhibit a fade-in/fade-out effect. Others show variations that are not accurately explained by the binary activity, such as footstep sounds on different surfaces.

In this chapter we propose the use of non-binary activity indicators to characterize the temporal activity of sound events: instead of estimating a point when a sound event becomes active/inactive, we propose to estimate its amplitude envelope. The estimation of the envelopes of the events within synthetic mixtures could provide a more representative modeling of their temporal activity. In the following sections, a novel approach to estimate acoustic mixtures envelopes by using convolutional recurrent neural networks (CRNNs) is described.

## 6.1 Sound event detection

Throughout this thesis, we have been focusing on the task of classifying sound events, assuming a pre-processing step in charge of isolating the sound events within the acoustic scene. As it has been seen, the way sound events are segmented affects considerably the performance of the classifier. In real-life environments, the amount of events and the number of categories present in the acoustic scene vary constantly, with the possibility of having events occurring at the same time, the so-called overlapping events. Sound event detection (SED) aims to detect presence of the different sounds in an audio recording and provide a textual label, onset and offset times for each [57]. In most SED-related tasks, sound events have to be detected in polyphonic mixtures, with either overlapping target sounds, or significant background present. Most state of the art methods use deep learning, with convolutional and recurrent neural networks being the most prominent [2, 64, 86, 82].

In order to train a model in a supervised manner to perform SED, we need ground-truth labels indicating the onsets and offsets of the audio events within the stream. These labels must be good descriptors of the different events and must have a clear correspondence with the sound they are representing. Using the strong labels provided by the annotators, the temporal boundaries of the sound events are used to objectively measure the level of accuracy of the trained model.

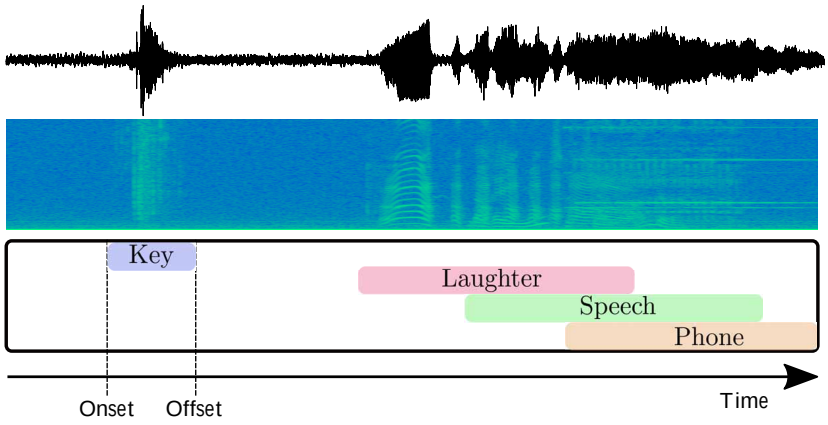



Figure 6.1: Annotating onset and offset times from a polyphonic mixture.

The provided human annotations have to be translated to the machine using a binary encoding in order to perform supervised training. This encoding means transforming the given annotations into a language that the machine can understand, by assigning "1" when the event is active and "0" otherwise. However, real-life acoustic scenes have a complicated behaviour. They can combine multiple events from the same class or from multiple different classes. Some of those sound events usually overlap, and while overlapping they may have different levels of intensity. This level of complexity is difficult to reflect by using only binary indicators, where a "1" would represent that a given sound is fully contained within that interval, while in reality it could be masked by a sound with a higher intensity. Therefore, more accurate indicators, able to reflect such behaviour, are needed. The problem of representing more than one class simultaneously, known as multi-label classification/detection has been tackled by having a matrix instead of a vector that indicates the presence of multiple classes. An example can be seen in Figure 6.1, where three different events occurring at the same time with their corresponding labels are represented. Once the annotations of all the classes are completed, all these information is encoded in a binary way using an standard approach like one-hot encoding.

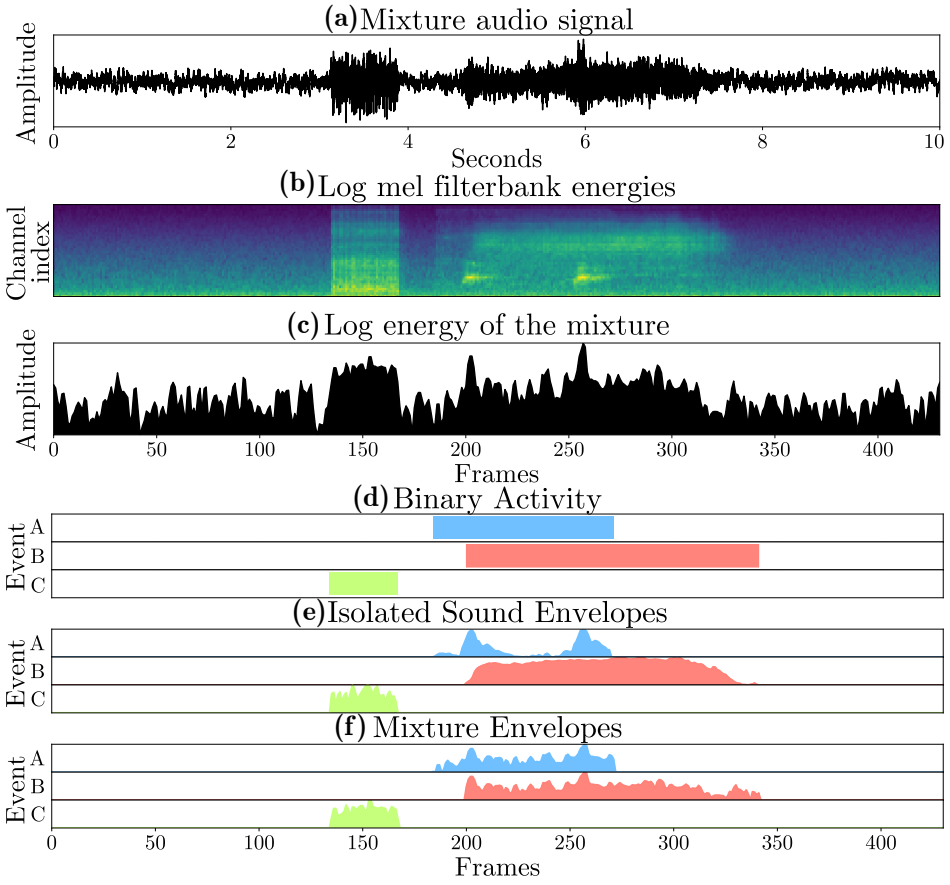



Figure 6.2: The process of obtaining envelopes for the isolated sounds and the mixtures based on the binary activity indicators.

As it has been mentioned, how well the encoded onset and offset timestamps are given to the system depends on how reliable the manual annotation is. Then, one way of obtaining reliable strong annotations for real sound mixtures without having to spend hours manually annotating such data, is to synthetically create those mixtures. That is what Salamon et al. did when they released the URBAN-SED database, collecting urban sounds from FreeSound to create mixtures using the Scaper tool [127], previously described in Section 2.6. By using this dataset as training data, the output of a system can be tested against the so-called gold ground-truth annotation in a straight-forward way.

### 6.1.1 Mixtures representation

In the previous Chapters 4 and 5, we have studied how using the energy of the sound events we can faithfully identify the important parts of the signal in terms of temporal activity. This idea motivated the use of a similar approach to accurately describe the structure of sound events within a mixture. For these experiments, the goal is to represent the acoustic mixtures avoiding binary labels to help the system better capture effects like the fade-in/fade-outs present in the signals of interest. Combining the information of the reference label and the energy extracted from the given mixture, a novel non-binary representation in the continuous domain can be defined, which will be referred here as soft-labeling. The process of calculating such representation is as follows:

- First of all, the log-mel spectrogram from a given mixture is calculated.

- From the extracted spectrogram the log energy of the mixture is obtained, known as the energy envelope of the mixture.
- The reference annotations are multiplied by the envelopes, obtaining the soft-labels.
- Finally, the obtained soft-labels are normalized between $[0, 1]$ and the logarithm is applied to have a smoother representation.

By using the log energy of the mixture, represented in Figure 6.2, gradual changes in the energy of the different sound events can be represented. These labels are used as a target to train the learning model.

The example in Figure 6.2 represents a mixture formed by three different sound events, with the corresponding binary activity indicators. Two kinds of soft-labels can be obtained from the information of the reference annotators. The isolated sound envelopes are calculated using the spectrogram from the individual sound events. As it can be seen in Figure 6.2 (e), the envelopes faithfully represent the overlapping sounds. However, in real-life data we do not have access to the isolated events. For this reason the (more realistic) envelopes extracted directly from the mixtures are also used as targets, depicted in Figure 6.2 (f). It can be seen how the mixture envelopes do not represent the sound events as faithfully as the isolated sound envelopes, however, they still give enough information to train the learning model, as will be shown in the next section.

### 6.1.2 Model design

The model architecture used in this work is a convolutional recurrent neural network (CRNN) based on the system proposed in [2], which ranked first for the "sound event detection in real-life audio" task in the DCASE 2017 challenge. As explained in Section 2.7, the CNNs are able to address the fixed connection limitation found in DNNs. However, they can not model long term temporal relationships within the signal. To tackle this problem, recurrent layers are added, given their ability of extracting temporal information from consecutive frames. An overview of the CRNN model is shown in Figure 6.3, where the three convolutional layers, each of them followed by batch normalization and max-pooling are depicted. The output of the CNN is fed to bi-directional gated recurrent units (GRU), which learn the temporal activity patterns. The last layers are time-distributed fully-connected (dense) layers. The output layer uses sigmoid activation to produce multiple continuous outputs. The input to the neural network consists of $T$ consecutive time frames of mel-band energies $N_{mbe}$; the dimensions are $T = 431$ given by the length of the audio files and $N_{mbe} = 40$ number of mel-bands in the frequency range of $0 - 22500Hz$.

To enable training with the continuous envelopes, the optimization loss function used has been set to the mean squared error (MSE) instead of the usual binary cross-entropy used for classification tasks. The best values for batch size and binarization threshold used to transform the regression output into detection are selected using the available validation set. The values we found worked best are batch size of 32 for mixtures, 16 for the isolated events; for both cases the best binarization threshold was 0.25. Training was performed using the usual Adam optimizer [67] with a learning rate of 0.001.

## 6.2 Experimental results

An example of the output from the trained regression model is depicted in Figure 6.4, where a mixture formed by two overlapping sound events (*dog bark* and *music*) is given as example.

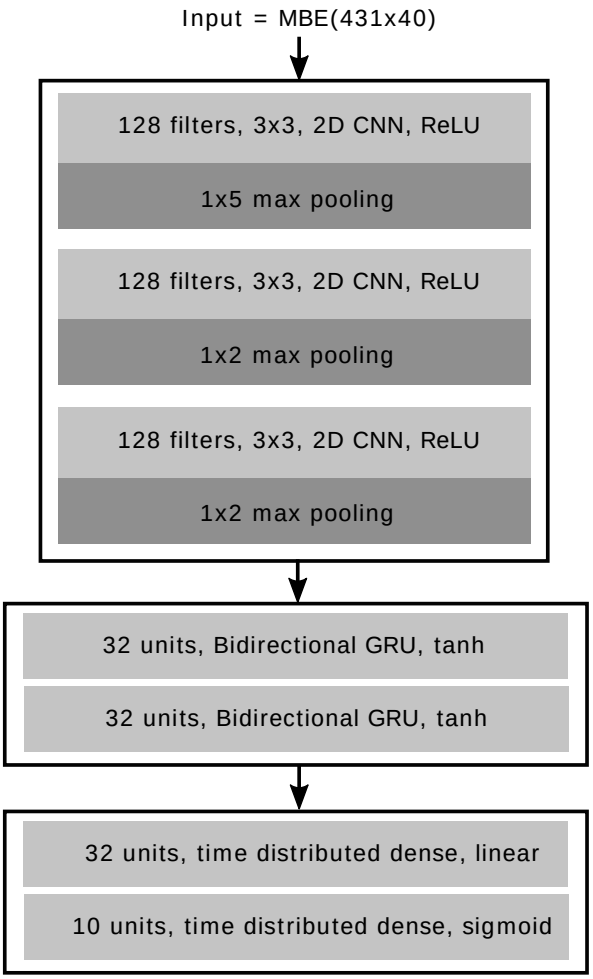



Figure 6.3: CRNN architecture used for sound event detection.

For this particular example the model has been trained using isolated envelopes. It can be seen that even though the predicted activity closely follows the ground truth, in some regions the event presence is marked by very low values.

In order to measure how well the model is able to estimate conditional envelopes from mixtures, the mean squared error (MSE) is calculated. Since absolute values of the MSE measure are difficult to interpret, the MSE of the resulting envelopes is calculated independently for each of two disjoint regions, active and inactive. However, the MSE obtained for inactive regions is not shown since it takes quite low values in the range of 0.002-0.009, meaning that the system is able to predict with high accuracy the truly inactive parts of all sound event classes. For the active regions on the other hand, we calculate the SNR of the estimated envelopes by dividing the energy of the reference envelopes ($Energy_{ref}$) by the corresponding squared error:

$$\mathrm{SNR} = 10 \log_{10} \left( \frac{Energy_{ref}}{Error} \right), \tag{6.1}$$

where $Error = \sum_{n=1}^{T} (ref[n] - pred[n])^2$, calculates the aggregated difference between the reference ($ref$) and predicted ($pred$) envelopes along time $T$.

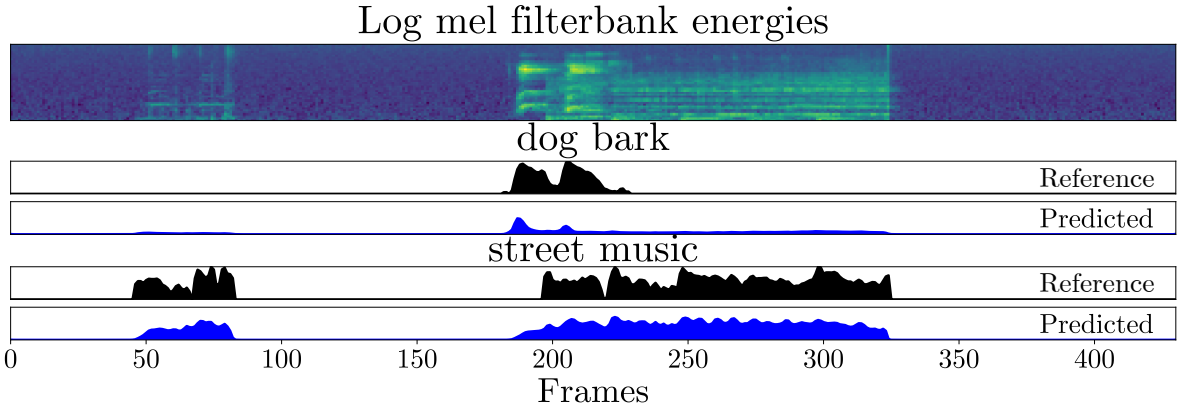



Figure 6.4: Envelopes estimated by the system trained with isolated sound envelopes.

| Event class | MSE | SNR [dB] |
|---|---|---|
| air conditioner | 0.190 | 2,658 |
| car horn | 0.181 | 3,459 |
| children playing | 0.152 | 3,198 |
| dog bark | 0.168 | 2,609 |
| drilling | 0.172 | 3,328 |
| engine idling | 0.148 | 3,917 |
| gun shot | 0.136 | 2,728 |
| jackhammer | 0.077 | 6,745 |
| siren | 0.129 | 4,187 |
| street music | 0.138 | 3,799 |

Table 6.1: Mean squared error of regression output and Signal to Noise Ratio (SNR) for active regions of the target sounds; training using mixture envelopes.

### Datasets

For this specific study that considers mixtures of events, we used the URBAN-SED dataset created using Scaper [126], as described in Section 2.6. Given the synthetic generation of the dataset, both hard and soft annotations are guaranteed to be correct and complete, compared to the uncertainty of other manually annotated datasets. The dataset also contains sound events such as dog barking and children playing that have fluctuating envelopes. In addition, this dataset allows us to verify our hypothesis that the events in the foreground can be represented using the mixture signal energy, by comparing the use of envelopes obtained from the original isolated sounds and those from the mixture signal as shown in Figure 6.2.

### 6.2.1 Envelope estimation results

Class-wise results obtained training the model using the envelopes from the mixtures, are given in Table 6.1, which includes both MSE and SNR from the active regions of the target sounds. From the MSE results it is difficult to evaluate what can be considered as a good

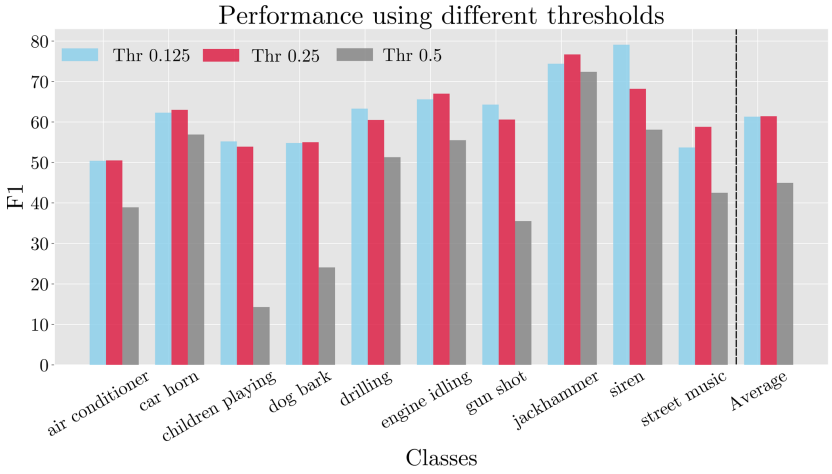



Figure 6.5: F1-score in 1 s segments for different binarization thresholds; training using envelopes from mixtures.

value. What we can conclude is that the system performs similarly for all the classes. Given the SNR values, we can differentiate how the system behaves among the distinct classes. For example, the classes jackhammer and siren, which are longer sound events, are the ones which are more accurately estimated. This result may be due to the higher intensity level of such classes compared to others (also long classes such as air conditioner). On the other hand, for shorter and impulsive classes like dog bark and gun shot, the system had problems at predicting their conditional envelopes.

If instead of using the mixture envelopes, the isolated envelopes are used for training the model, then it is possible to improve the estimation of such short classes. Moreover, the remaining classes are not affected by this change. We can conclude that, even by using mixture envelopes for training (which can be considered as a more realistic scenario) the system is able to successfully identify the non-active parts of the mixtures, while at the same time, it is also capable of obtaining a close representation of the active parts, similarly as with the idealistic isolated envelopes.

### 6.2.2 Sound event detection results

To perform a more precise analysis of the system capabilities to estimate envelopes from mixtures, the problem has been reformulated as a SED task. Then, the predicted output from the trained regression model is thresholded using a trade-off value, to obtain binary indicators. Once the binary labels are obtained, the classification performance is calculated using a segment-based approach, as described in Section 2.4. In particular, both the Error Rate and the F-score within 1 second segment length are obtained.

After thresholding the estimated envelopes, a post-process is done in order to discard some errors due to the high variability of the sound events. The very short segments separated within a certain gap are joined into a bigger event while the segments above that gap are separated to form new ones. Finally, the ones below a minimum length are discarded, since it is likely that they are caused by some sounds in the background.

The results corresponding to the SED task show how the value chosen for binarization highly affects the final classification performance. A traditional value used for thresholding the model output is 0.5, however this value has shown to report low performance results in

| Event class | binary | isolated env. | mixture env. |
|---|---|---|---|
| air conditioner | 48.3 | 49.2 | 50.5 |
| car horn | 66.0 | 66.9 | 63.0 |
| children playing | 56.9 | 56.7 | 53.9 |
| dog bark | 60.3 | 59.6 | 55.0 |
| drilling | 66.3 | 63.0 | 60.5 |
| engine idling | 68.2 | 67.0 | 67.0 |
| gun shot | 71.5 | 60.7 | 60.6 |
| jackhammer | 78.3 | 78.6 | 76.7 |
| siren | 69.9 | 69.0 | 68.2 |
| street music | 59.7 | 60.5 | 58.8 |
| average | 64.3 | 63.1 | 61.4 |

Table 6.2: F1-score in 1 s segments for different approaches to detection; estimated envelopes binarized with 0.25 threshold.

| System training | F1 | ER |
|---|---|---|
| binary activity | 64.7 | 0.48 |
| envelope from isolated examples | 63.6 | 0.49 |
| envelope from mixture signal | 61.8 | 0.52 |

Table 6.3: F1-score and error rate calculated using micro-averaging (1 s segment-based).

our case. Different values of thresholding have been tested, as depicted in Figure 6.5, where the average values for thresholds 0.125 and 0.25 are very close (61.31 vs 61.42). Based on the validation data, the threshold of 0.25 was finally selected as the one leading to best error rate (ER) and F1-score.

Table 6.2 shows the F1-score values for each of the classes, comparing the training performed using binary indicators, isolated envelopes and mixture envelopes. For the binary indicators the threshold is set at 0.5 while for the other two scenarios it has been changed to 0.25. The overall performance is better for the system using binary information, which obtains state-of-the-art performance, with even higher values than the ones reported in [126]. In the case of short sound events, as it has been shown, the system is not able to properly detect such short activities, with a degradation close to 10% for the gun-shot class. Since very short events in the regression output are filtered out by the post-processing, it may also be the case that some detected very short gunshot events are discarded.

The detection results using ER and F1-score as used in the DCASE Challenge were also evaluated. In Table 6.2, the overall accumulation of counts before metric calculation using micro-averaging is given. The difference between the class-wise results is rather small, since the system performance is consistent between classes and the dataset is quite balanced. The

presented results show that estimating the sound conditional envelopes provides SED results which are comparable with the state-of-the-art performance apart from obtaining enriched information about the local activity pattern of each event.

## 6.3 Conclusion

This chapter presented a novel regression-based sound event detection method based on soft activity labels computed from the short-time energy of the mel-spectrograms of the audio events. Although the performance using binary activity indicators was slightly better than the one obtained by the proposed soft-detection scheme, the results achieved by such system can be considered promising and motivate further research in this direction. In fact, the conditional envelope estimation results, evaluated in terms of mean squared error and SNR, reflect the effectiveness of the proposed method. After transforming the envelopes into activity descriptors, the detection capability of the proposed method has been shown to lead to performance results comparable to state-of-the-art systems trained using traditional binary activity indicators.

This method could be of great interest since it can be also applied for alleviating the subjectivity present due to human annotators when labeling the time boundaries of sound events in complex mixtures. Instead of having strict annotations, the uncertainty can be represented using a non-binary activity score, which in turn may help to accurately describe the temporal structure of some sounds.

# Chapter 7

# Conclusions

Applications related to the automatic understanding of the sound environment have been receiving increasing attention in recent years. Compared to speech recognition, which is a more mature field and one of the most active research domains in machine learning, scientific work dealing with general sound event recognition tasks has increased exponentially. The growing interest is mainly motivated by the elevated number of potential industrial applications: voice assistants, smart hearing aids, sound indexing and retrieval, bioacoustics, predictive maintenance, ambient assisted living or security.

In this thesis, we have addressed the problem of robust sound event recognition considering both the two-step classical approach (using feature extraction followed by classification) and recent frameworks based on convolutional neural networks. Machine listening systems for recognizing acoustic events share some similarities with humans. First, the systems need to learn discriminative patterns for different classes in a supervised manner by using labeled data. Once such patterns have been learned, the system is then expected to correctly identify new unknown examples, predicting a class name for a given input. However, different problems may appear in the process. On the one hand, it is not always possible to rely on the available training data. This data may be corrupted, noisy, or insufficient to be trustfully employed for recognizing some of the classes of interest. On the other hand, even when the training data meets the required quality standards both in terms of content and annotation, it is not rare that the system is later tested in acoustic environments completely different from those where the training examples were recorded.

In Chapter 2, we provided an overview of the most common sound event recognition tasks. The general audio processing steps needed within the recognition pipeline were introduced. The way data are presented to a classifier is an important issue affecting the performance of the system and, at the same time, depends on the task at hand. To create a meaningful representation, features have to be properly selected, assuring that they are contributing to enhance the most notable differences among different classes. This chapter also introduces the different evaluation metrics, machine learning models and datasets used throughout the thesis. Finally, common and open problems related to sound event recognition tasks are introduced, some of which are addressed in the following chapters.

The performance provided by hand-crafted and deep features in terms of robustness to acoustic mismatches between the training and testing sets was elaborated in Chapter 3. First, the sensitivity of common sets of hand-crafted features used in general sound recognition tasks was analyzed in this chapter by means of a pruning-based algorithm. Then,

the influence on the performance due to changes in background noise and reverberation at test time was discussed both for hand-crafted and deep features. The results obtained in this context, suggest that, mel-frequency energies tend to be more informative than MFCCs for sound event classification tasks. This is in agreement with the choice taken by current state-of-the-art deep learning systems, which are widely using log-mel spectrograms as inputs. Additionally, the performance degradation caused by mismatching acoustic conditions was confirmed, observing that while classical and deep features are substantially affected by noise and reverberation, deep features seem to be more robust to background noise.

Chapter 4 presented a new method for transforming features extracted from variable-length input sequences into fixed-length representations embedding the temporal activities of acoustic events. The method was designed to be robust against weakly segmented audio events and background noise, two effects that appear frequently in realistic sound event recognition applications. Specifically, we proposed the use of a non-linear time normalization of short-time features by considering a uniform distance subsampling criterion over an alternative low-dimensionality feature space different from the one used for classification. The so-called trace, which refers to the accumulated distance of time consecutive points in the feature space, was shown to be a very useful tool for capturing the temporal activity of the events. The experiments confirmed the utility of such transformation, which improves the classification performance under a variety of adverse scenarios.

Given the promising results obtained by the proposed trace-based transformation on hand-crafted features, the same principles were adapted to a fully end-to-end deep learning framework in Chapter 5. This was conducted by proposing a novel distance-based pooling layer for convolutional neural networks that incorporates activity information within the learning process. In this case, the activations from an internal layer of the network are non-linearly downsampled on the temporal axis guided by a uniform segmentation of the computed trace. The experiments performed over a baseline state-of-the-art recognition system showed that including the proposed pooling scheme made the system more robust to weakly segmented test data.

The problem of sound event detection (different from event classification), was finally addressed in Chapter 6. A novel method for training a deep detection model using the short-term energies of acoustic events as target ground-truths was proposed. In contrast to most recognition systems, where the detection task is approached from a classification perspective, the problem was here formulated from a regression point of view. This in turn led to a new training scheme, where instead of using binary activations as targets, the system is fed with the temporal activity given by the short-term log-mel energy of the signals. Such method has shown promising results, comparable to that of current state-of-the-art systems.

In summary, the main results of this thesis have made evident the importance of having robust data representations in the development of powerful sound event recognition systems exposed to adverse scenarios. In this context, the energy envelope of audio events has been shown to be very helpful to incorporate temporal activity information within the learning phase of machine listening models. Indeed, whether classical approaches or deep learning frameworks are used, dealing with the variable-length nature of audio events is still a challenging issue.

## 7.1 Further work

The research work carried out in this thesis has evidenced that there are still many challenges to be faced in the future regarding the design of new sound recognition systems working in a variety of environments and real-life situations.

Some of these challenges are summarized as follows:

- The cost related to obtaining huge amounts of training data with reliable annotations motivates to look further into training methods dealing more efficiently with weak labels. This thesis has particularly addressed the problem of sound event classification of weakly segmented data, which is closely related to weak labeling. The similarities shared among both problems should be closely explored so that future research could be oriented towards this direction.

- In relation to the above problem, the uncertainty present in the timestamps marked by human annotators may be quite relevant. There are no studies discussing the effect of this uncertainty and, so far, it is unknown if it affects the learning process. Consideration of this temporal uncertainty may have great potential to make recognition systems more robust to adverse conditions present in real-life scenarios and, consequently, to improve their performance.

- The idea of using the trace, introduced in Chapter 4 and transferred to deep models in Chapter 5, has led to robust models in our particular problem. But there is path ahead to develop this strategy and arrive at new general adaptive or specialized pooling operations that may play a role similar to dropout in order to boost robustness and tolerance of the whole system in any data domain.

- Finally, although not addressed in this thesis, the open-set recognition problem is of major interest for developing new applications working on a general real-life environment, where the systems may be challenged with lots of instances coming from multiple unexpected classes. While some approaches have been proposed in the literature using traditional one-class classifiers, further work is needed to obtain solutions making use of the full potential of deep models.

## 7.2 Publications

The main results and contributions of this thesis have been published in the following journal articles and conference papers.

### Journal publications

**Publication 1:**

I. Martín-Morató, M. Cobos and F. J. Ferri, "Adaptive mid-term representations for robust audio event classification," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 12, pp. 2381-2392, Dec. 2018.
doi: 10.1109/TASLP.2018.2865615

This publication presents the contributions described in Chapter 4 of this thesis, related to the trace-based transformation of short-term features to embed the temporal activity of

audio events into fixed-length feature vectors.

**Publication 2:**

I. Martín-Morató, M. Cobos and F. J. Ferri, "Distance-based pooling in convolutional neural networks for audio event classification," submitted to *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2019.

The above paper contains the proposal of the adaptive pooling layer of Chapter 5, which adapts the trace-based tranformation idea to a deep learning framework.

### International conferences

**Publication 3:**

I. Martín-Morató, M. Cobos and F. J. Ferri,"A case study on feature sensitivity for audio event classification using support vector machines," *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, Vietri sul Mare, 2016, pp. 1-6.
doi: 10.1109/MLSP.2016.7738834

This contribution presents part of the results in Chapter 3, related to the sensitivity analysis of hand-crafted features.

**Publication 4:**

I. Martín-Morató, M. Cobos and F. J. Ferri, "Analysis of data fusion techniques for multi-microphone audio event detection in adverse environments," *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, Luton, 2017, pp. 1-6.
doi: 10.1109/MMSP.2017.8122274

This publication elaborates on the robustness analysis of hand-crafted features in realistic scenarios considering fusion techniques, as described in Chapter 3.

**Publication 5:**

I. Martin-Morato, M. Cobos and F. J. Ferri, "On the robustness of deep features for audio event classification in adverse environments," *2018 14th IEEE International Conference on Signal Processing (ICSP)*, Beijing, China, 2018, pp. 562-566.
doi: 10.1109/ICSP.2018.8652438

The above conference paper presents the experiments carried out in Chapter 3 to assess the robustness of deep features in adverse scenarios.

**Publication 6:**

I. Martín-Morató, A. Mesaros, T. Heittola, T. Virtanen, M. Cobos and F. J. Ferri, "Sound Event Envelope Estimation in Polyphonic Mixtures," *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2019)* , Brighton, United Kingdom, 2019, pp. 935-939.
doi: 10.1109/ICASSP.2019.8682858

This publication presents the regression-based detection system based on the estimation of event envelopes presented in Chapter 6.

**Publication 7:**

I. Martin-Morato, M. Cobos and F. J. Ferri, "Performance analysis of audio event classification using deep features under adverse acoustic conditions," in *Proceedings of the 23rd International Congress on Acoustics (ICA 2019)*, Aachen, Germany, 2019, pp. 6980 - 6987.

This publication extends the analysis of the robustness of deep features presented in Chapter 3 by evaluating the performance of features extracted from different network depths.

# Bibliography


[1] C. Nadeu C. Segura A. Temko, D. Macho. Upc-talp database of isolated acoustic events. In *Internal UPC report*, 2005.

[2] S. Adavanne, P. Pertilä, and T. Virtanen. Sound event detection using spatial features and convolutional recurrent neural network. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 771–775, March 2017.

[3] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional space. In Jan Van den Bussche and Victor Vianu, editors, *Database Theory — ICDT: 8th International Conference London, UK, January 4–6, Proceedings*, pages 420–434. Springer Berlin Heidelberg, 2001.

[4] J. B. Allen and D. A. Berkley. Image method for efficiently simulating small-room acoustics. *J. Acoust. Soc. Amer.*, 65(3):943–950, 1979.

[5] Sven E. Anderson, Amish S. Dave, and Daniel Margoliash. Template-based automatic recognition of birdsong syllables from continuous recordings. *Journal of the Acoustical Society of America*, 100:1209–1219, 1996.

[6] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals. Speaker diarization: A review of recent research. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2):356–370, Feb 2012.

[7] R. Anitha, D. S. Satish, and C. C. Sekhar. Outerproduct of trajectory matrix for acoustic modeling using support vector machines. In *Proceedings of the 14th IEEE Signal Processing Society Workshop MLSP*, pages 355–363, 2004.

[8] P. Arora and R. Haeb-Umbach. A study on transfer learning for acoustic event detection in a real life scenario. In *IEEE 19th MMSP*, pages 1–6, Oct 2017.

[9] A.S. Bregman. *Auditory scene analysis: The perceptual organization of sound.* The MIT Press, 1990.

[10] Pradeep K. Atrey, M. Anwar Hossain, Abdulmotaleb El-Saddik, and Mohan S. Kankanhalli. Multimodal fusion for multimedia analysis: a survey. *Multimedia Syst.*, 16(6):345–379, 2010.

[11] Jean-Julien Aucouturier, Boris Defreville, and François Pachet. The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music. *The Journal of the Acoustical Society of America*, 122:881, 2007.

[12] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. In *Advances in Neural Information Processing Systems*, 2016.

[13] D. Battaglino, L. Lepauloux, and N. Evans. The open-set problem in acoustic scene classification. In *2016 IEEE International Workshop on Acoustic Signal Enhancement (IWAENC)*, pages 1–5, Sep. 2016.

[14] Daniela Beltrami, Gloria Gagliardi, Rema Rossini Favretti, Enrico Ghidoni, Fabio Tamburini, and Laura Calzà. Speech analysis by natural language processing techniques: A possible tool for very early detection of cognitive decline? *Frontiers in Aging Neuroscience*, 10:369, 2018.

[15] Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri. Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems*, 41(3):407–434, Dec 2013.

[16] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, Aug 2013.

[17] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[18] T. Bocklet, A. Maier, J. G. Bauer, F. Burkhardt, and E. Noth. Age and gender recognition for telephone applications based on gmm supervectors and support vector machines. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1605–1608, March 2008.

[19] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA, 1992. ACM.

[20] A.L. Brown, Jian Kang, and Truls Gjestland. Towards standardization in soundscape preference assessment. *Applied Acoustics*, 72(6):387 – 392, 2011.

[21] William E. Brownell and Paul B. Manis. Structures, mechanisms, and energetics in temporal processing. In Arthur N. Popper and Richard R. Fay, editors, *Perspectives on Auditory Research*, pages 9–44. Springer New York, New York, NY, 2014.

[22] M. Omologo C. Zieger. Acoustic event detection - itc-irst aed database. In *Internal ITC report*, 2005.

[23] E. F. Cabral and G. D. Tattersall. Trace-segmentation of isolated utterances for speech recognition. In *International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 365–368, 1995.

[24] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen. Polyphonic sound event detection using multi label deep neural networks. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, July 2015.

[25] M. J. Carey, E. S. Parris, and H. Lloyd-Thomas. A comparison of features for speech, music discrimination. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*, volume 1, pages 149–152 vol.1, March 1999.

[26] M. Casey. Mpeg-7 sound-recognition tools. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):737–747, June 2001.

[27] E. C. Cherry. Some Experiments on the Recognition of Speech, with One and with Two Ears. *Acoustical Society of America Journal*, 25:975, 1953.

[28] Maximo Cobos, Fabio Antonacci, Anastasios Alexandridis, Athanasios Mouchtaris, and Bowon Lee. A survey of sound source localization methods in wireless acoustic sensor networks. *Wireless Communications and Mobile Computing*, 2017(Article ID 3956282):1–24, 2017.

[29] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995.

[30] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, August 1980.

[31] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, May 2011.

[32] J. Dennis, H. D. Tran, and H. Li. Spectrogram image feature for sound event classification in mismatched conditions. *IEEE Signal Processing Letters*, 18(2):130–133, Feb 2011.

[33] J. Dennis, H. D. Tran, and H. Li. Combining robust spike coding with spiking neural networks for sound event classification. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 176–180, April 2015.

[34] Wladimiro Díaz-Villanueva, Francesc J. Ferri, and Vicente Cerverón. Learning improved feature rankings through decremental input pruning for support vector based drug activity prediction. In Nicolás García-Pedrajas, Francisco Herrera, Colin Fyfe, José Manuel Benítez, and Moonis Ali, editors, *Trends in Applied Intelligent Systems*, pages 653–661, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[35] S. Dieleman and B. Schrauwen. End-to-end learning for music audio. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6964–6968, May 2014.

[36] Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *J. Artif. Int. Res.*, 2(1):263–286, January 1995.

[37] A. Diment and T. Virtanen. Transfer learning of weakly labelled audio. In *IEEE WASPAA*, pages 6–10, Oct 2017.

[38] Ke-Lin Du and M. N.S. Swamy. *Neural Networks and Statistical Learning*. Springer Publishing Company, Incorporated, 2013.

[39] H. Eghbal-zadeh, B. Lehner, M. Dorfer, and G. Widmer. A hybrid approach with multi-channel i-vectors and convolutional neural networks for acoustic scene classification. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 2749–2753, Aug 2017.

[40] K. El-Maleh, M. Klein, G. Petrucci, and P. Kabal. Speech/music discrimination for multimedia applications. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, volume 4, pages 2445–2448 vol.4, June 2000.

[41] Tuomas Virtanen Emre Çakır. Convolutional recurrent neural networks for rare sound event detection. In *Proceedings of the Detection and classification of Acoustic Scenes and Events (DCASE)*, pages 27–31, November 2017.

[42] A. J. Eronen, V. T. Peltonen, J. T. Tuomi, A. P. Klapuri, S. Fagerlund, T. Sorsa, G. Lorho, and J. Huopaniemi. Audio-based context recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):321–329, Jan 2006.

[43] Eduardo Fonseca, Manoj Plakal, Daniel P. W. Ellis, Frederic Font, Xavier Favory, and Xavier Serra. Learning sound event classifiers from web audio with noisy labels. *CoRR*, abs/1901.01189, 2019.

[44] Frederic Font, Gerard Roma, and Xavier Serra. Freesound technical demo. In *ACM International Conference on Multimedia (MM'13)*, pages 411–412, Barcelona, Spain, 21/10/2013 2013. ACM, ACM.

[45] R. Foucard, J. Durrieu, M. Lagrange, and G. Richard. Multimodal similarity between musical streams for cover version detection. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5514–5517, March 2010.

[46] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[47] David Gerhard. Audio signal classification: History and current techniques. Technical report, A. TEMKO, C. NADEU / PATTERN RECOGNITION 39 (2006) 682 – 694, 2003.

[48] S. Gharib, H. Derrar, D. Niizumi, T. Senttula, J. Tommola, T. Heittola, T. Virtanen, and H. Huttunen. Acoustic scene classification: a competition review. In *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, Sep. 2018.

[49] T. Giannakopoulos, S. Petridis, and S. Perantonis. User-driven recognition of audio events in news videos. In *2010 Fifth International Workshop Semantic Media Adaptation and Personalization*, pages 44–49, Dec 2010.

[50] Theodoros Giannakopoulos and Aggelos Pikrakis. Chapter 4 - audio features. In *Introduction to Audio Analysis*, pages 59 – 103. Academic Press, Oxford, 2014.

[51] P. Giannoulis, G. Potamianos, A. Katsamanis, and P. Maragos. Multi-microphone fusion for detection of speech and acoustic events in smart spaces. In *2014 22nd European Signal Processing Conference (EUSIPCO)*, pages 2375–2379, Sept 2014.

[52] A. Graves, A. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649, May 2013.

[53] Alex Graves and Jürgen Schmidhuber. Offline handwriting recognition with multidimensional recurrent neural networks. In *Proceedings of the 21st International Conference on Neural Information Processing Systems*, NIPS'08, pages 545–552, USA, 2008. Curran Associates Inc.

[54] Catherine Guastavino. Everyday sound categorization. In Tuomas Virtanen, Mark D. Plumbley, and Dan Ellis, editors, *Computational Analysis of Sound Scenes and Events*, pages 183–213. Springer International Publishing, Cham, 2018.

[55] Guodong Guo and S. Z. Li. Content-based audio classification and retrieval by support vector machines. *IEEE Transactions on Neural Networks*, 14(1):209–215, Jan 2003.

[56] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[57] T. Heittola, E. Çakır, and T. Virtanen. The machine learning approach for analysis of sound scenes and events. In Tuomas Virtanen, Mark D. Plumbley, and Dan Ellis, editors, *Computational Analysis of Sound Scenes and Events*, pages 13–40. Springer International Publishing, Cham, 2018.

[58] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson. Cnn architectures for large-scale audio classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 131–135, March 2017.

[59] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[60] Y. Hoshen, R. J. Weiss, and K. W. Wilson. Speech acoustic modeling from raw multichannel waveforms. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4624–4628, April 2015.

[61] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160(1):106–154, 1962.

[62] Acoustics – Soundscape – Part 1: Definition and conceptual framework. Standard, International Organization for Standardization, Geneva, CH, 2014.

[63] Anil Jain and Douglas Zongker. Feature selection: Evaluation, application, and small sample performance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(2):153–158, 1997.

[64] I-Y. Jeong, S. Lee, Y. Han, and K. Lee. Audio event detection using multiple-input convolutional neural network. In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, pages 51–54, November 2017.

[65] Burred Juan José Kim, Hyoung-Gook and Thomas Sikora. How efficient is mpeg-7 for general sound recognition? In *Audio Engineering Society Conference: 25th International Conference: Metadata for Audio*, Jun 2004.

[66] T. Kim, J. Lee, and J. Nam. Sample-level cnn architectures for music auto-tagging using raw waveforms. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 366–370, April 2018.

[67] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[68] Josef Kittler, Mohamad Hatef, Robert P. W. Duin, and Jiri Matas. On combining classifiers. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(3):226–239, 1998.

[69] Q. Kong, Y. Xu, I. Sobieraj, W. Wang, and M. D. Plumbley. Sound event detection and time–frequency segmentation from weakly labelled data. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(4):777–787, April 2019.

[70] Q. Kong, Y. Xu, W. Wang, and M. D. Plumbley. A joint detection-classification model for audio tagging of weakly labelled data. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 641–645, March 2017.

[71] Zvi Kons and Orith Toledo-Ronen. Audio event classification using deep neural networks. In *INTERSPEECH*, 2013.

[72] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[73] M. Kuhn, H. Tomaschewski, and H. Ney. Fast nonlinear time alignment for isolated word recognition. In *ICASSP IEEE*, volume 6, pages 736–740, 1981.

[74] Anurag Kumar and Bhiksha Raj. Audio event detection using weakly labeled data. In *Proceedings of the 24th ACM International Conference on Multimedia*, MM '16, pages 1038–1047, New York, NY, USA, 2016. ACM.

[75] Anurag Kumar and Bhiksha Raj. Deep CNN framework for audio event recognition using weakly labeled web data. *CoRR*, abs/1707.02530, 2017.

[76] G. Lafay, E. Benetos, and M. Lagrange. Sound event detection in synthetic audio: Analysis of the dcase 2016 task results. In *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 11–15, Oct 2017.

[77] G. Lafay, M. Lagrange, M. Rossignol, E. Benetos, and A. Roebel. A morphological model for simulating acoustic scenes and its application to sound event detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(10):1854–1864, Oct 2016.

[78] Mathieu Lagrange. Simscene: simulation of acoustic scenes. `https://bitbucket.org/mlagrange/simscene/downloads/`, 2018.

[79] Brenden M Lake, Ruslan R Salakhutdinov, and Josh Tenenbaum. One-shot learning by inverting a compositional causal process. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2526–2534. Curran Associates, Inc., 2013.

[80] S. Lecomte, R. Lengellé, C. Richard, F. Capman, and B. Ravera. Abnormal events detection using unsupervised one-class svm - application to audio surveillance and evaluation. In *2011 8th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 124–129, Aug 2011.

[81] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, Dec 1989.

[82] D. Lee, S. Lee, Y. Han, and K. Lee. Ensemble of convolutional neural networks for weakly-supervised sound event detection using multiple scale input. In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, pages 74–79, 2017.

[83] Jongpil Lee, Taejun Kim, Jiyoung Park, and Juhan Nam. Raw waveform-based audio classification using sample-level cnn architectures. In *Neural Information Processing Systems (NIPS)*, 12 2017.

[84] Luis A. Leiva and Enrique Vidal. Warped k-means: An algorithm to cluster sequentially-distributed data. *Information Sciences*, 237:196 – 210, 2013.

[85] Sarah Ita Levitan, Taniya Mishra, and Srinivas Bangalore. Automatic identification of gender from speech. In *Speech Prosody 2016*, pages 84–88, 2016.

[86] H. Lim, J. Park, and Y. Han. Rare sound event detection using 1D convolutional recurrent neural networks. In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*, pages 80–84, November 2017.

[87] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. In *International Conference on Learning Representations (ICLR)*, 2014.

[88] David Little and Bryan Pardo. Learning musical instruments from mixtures of audio with weak labels. In *In Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR*, 2008.

[89] Huan Liu, Hiroshi Motoda, Rudy Setiono, and Zheng Zhao. Feature selection: An ever evolving frontier in data mining. In Huan Liu, Hiroshi Motoda, Rudy Setiono, and Zheng Zhao, editors, *Proceedings of the Fourth International Workshop on Feature Selection in Data Mining*, volume 10 of *Proceedings of Machine Learning Research*, pages 4–13, Hyderabad, India, 21 Jun 2010. PMLR.

[90] S. Liu and W. Deng. Very deep convolutional neural network based image classification using small training sample size. In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 730–734, Nov 2015.

[91] Beth Logan. Mel frequency cepstral coefficients for music modeling. In *In International Symposium on Music Information Retrieval*, 2000.

[92] M.M. Marcell, D. Borella, M. Greene, E. Kerr, and S. Rogers. Confrontation naming of environmental sounds. *Journal of Clinical and Experimental Neuropsychology*, 22(6):830–864, 2000. cited By 109.

[93] I. Martín-Morató, M. Cobos, and F. J. Ferri. A case study on feature sensitivity for audio event classification using support vector machines. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, Sep. 2016.

[94] I. Martín-Morató, M. Cobos, and F. J. Ferri. Analysis of data fusion techniques for multi-microphone audio event detection in adverse environments. In *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6, Oct 2017.

[95] I. Martín-Morató, M. Cobos, and F. J. Ferri. Adaptive mid-term representations for robust audio event classification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(12):2381–2392, Dec 2018.

[96] I. Martín-Morató, M. Cobos, and F. J. Ferri. On the robustness of deep features for audio event classification in adverse environments. In *2018 14th IEEE International Conference on Signal Processing (ICSP)*, pages 562–566, Aug 2018.

[97] I. Martín-Morató, A. Mesaros, T. Heittola, T. Virtanen, M. Cobos, and F. J. Ferri. Sound event envelope estimation in polyphonic mixtures. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 935–939, May 2019.

[98] A. Mesaros, T. Heittola, E. Benetos, P. Foster, M. Lagrange, T. Virtanen, and M. D. Plumbley. Detection and classification of acoustic scenes and events: Outcome of the dcase 2016 challenge. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(2):379–393, Feb 2018.

[99] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen. Acoustic event detection in real life recordings. In *2010 18th European Signal Processing Conference*, pages 1267–1271, Aug 2010.

[100] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. Metrics for polyphonic sound event detection. *Applied Sciences*, 6(6), 2016.

[101] Ha Quang Minh, Partha Niyogi, and Yuan Yao. Mercer's theorem, feature maps, and smoothing. In Gábor Lugosi and Hans Ulrich Simon, editors, *Learning Theory: 19th Annual Conference on Learning Theory, COLT 2006, Pittsburgh, PA, USA, June 22-25, 2006. Proceedings*, pages 154–168, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[102] J.D. Krijnders M.W.W. Grootel, T.C. Andringa. DARES-G1: Database of Annotated Real-world Everyday Sounds. In *Proceedings of the NAG/DAGA Meeting 2009*, Rotterdam, 2009.

[103] Satoshi Nakamura, Kazuo Hiyane, Futoshi Asano, and Takanobu Nishiura. Acoustical sound database in real environments for sound scene understanding and hands-free speech recognition. In *Proceedings of 2nd ICLRE*, pages 965–968, 2000.

[104] Javier Naranjo-Alcazar, Sergi Perez-Castanos, Irene Martín-Morató, Pedro Zuccarello, and Maximo Cobos. On the performance of residual block design alternatives in convolutional neural networks for end-to-end audio classification. *CoRR*, abs/1906.10891, 2019.

[105] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio, 2016.

[106] Sergio Oramas, Oriol Nieto, Francesco Barbieri, and Xavier Serra. Multi-label music genre classification from audio, text and images using deep features. In *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, pages 23–30, 2017.

[107] D. Palaz, M. Magimai.-Doss, and R. Collobert. Convolutional neural networks-based continuous speech recognition using raw speech signal. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4295–4299, April 2015.

[108] K. K. Paliwal. On the use of filter-bank energies as features for robust speech recognition. In *ISSPA '99. Proceedings of the Fifth International Symposium on Signal Processing and its Applications (IEEE Cat. No.99EX359)*, volume 2, pages 641–644 vol.2, Aug 1999.

[109] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, Oct 2010.

[110] James W. Perry, Allen Kent, and Madeline M. Berry. Machine literature searching x. machine language; factors underlying its design and development. *American Documentation*, 6(4):242–254, 1955.

[111] Sergios Petridis, Theodoros Giannakopoulos, and Stavros Perantonis. A multi-class method for detecting audio events in news broadcasts. In Stasinos Konstantopoulos, Stavros Perantonis, Vangelis Karkaletsis, Constantine D. Spyropoulos, and George Vouros, editors, *Artificial Intelligence: Theories, Models and Applications: 6th Hellenic Conference on AI, SETN 2010, Athens, Greece, May 4-7, 2010. Proceedings*, pages 399–404. Springer Berlin Heidelberg, 2010.

[112] Silvia Pfeiffer. Pause concepts for audio segmentation at different semantic levels. In *Proceedings of the Ninth ACM International Conference on Multimedia*, MULTIMEDIA '01, pages 187–193, New York, NY, USA, 2001. ACM.

[113] H. Phan, P. Koch, F. Katzberg, M. Maass, R. Mazur, I. McLoughlin, and A. Mertins. What makes audio event detection harder than classification? In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 2739–2743, Aug 2017.

[114] K. J. Piczak. Environmental sound classification with convolutional neural networks. In *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, Sep. 2015.

[115] Karol J. Piczak. Esc: Dataset for environmental sound classification. In *ACM Multimedia*, 2015.

[116] John C. Platt. Fast training of support vector machines using sequential minimal optimization. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, *Advances in Kernel Methods*, pages 185–208. MIT Press, Cambridge, MA, USA, 1999.

[117] A. Plinge, R. Grzeszick, and G. A. Fink. A bag-of-features approach to acoustic event detection. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3704–3708, May 2014.

[118] J. Pons, J. Serrà, and X. Serra. Training neural audio classifiers with few data. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 16–20, May 2019.

[119] Jordi Pons, Oriol Nieto, Matthew Prockup, Erik M. Schmidt, Andreas F. Ehmann, and Xavier Serra. End-to-end learning for music audio tagging at scale. In Emilia Gómez, Xiao Hu, Eric Humphrey, and Emmanouil Benetos, editors, *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*, pages 637–644, 11 2018.

[120] A. Rabaoui, M. Davy, S. Rossignol, and N. Ellouze. Using one-class svms and wavelets for audio surveillance. *IEEE Transactions on Information Forensics and Security*, 3:763–775, Dec 2008.

[121] C. J. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979.

[122] Curtis Roads. *The Computer Music Tutorial*. MIT Press, Cambridge, MA, USA, 1996.

[123] David E. Rumelhart, De Rumerhart, Geoffrey E. Hinton, and Richard J. Williams. Learning representations of back-propagation errors. *Nature*, (323):533–536, 1986.

[124] Tara Sainath, Ron J. Weiss, Kevin Wilson, Andrew W. Senior, and Oriol Vinyals. Learning the speech front-end with raw waveform cldnns. In *Interspeech*, 2015.

[125] J. Salamon and J. P. Bello. Unsupervised feature learning for urban sound classification. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 171–175, April 2015.

[126] J. Salamon and J. P. Bello. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24(3):279–283, March 2017.

[127] J. Salamon, D. MacConnell, M. Cartwright, P. Li, and J. P. Bello. Scaper: A library for soundscape synthesis and augmentation. In *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 344–348, Oct 2017.

[128] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. A dataset and taxonomy for urban sound research. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 1041–1044, New York, NY, USA, 2014. ACM.

[129] Chris Sanden and John Z. Zhang. Enhancing multi-label music genre classification through ensemble techniques. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 705–714, New York, NY, USA, 2011. ACM.

[130] R. Murray Schafer. *The Tuning of the World: Toward a Theory of Soundscape Design.* University of Pennsylvania Press, 1977.

[131] M. Schedl, E. Gómez, and J. Urbano. *Music Information Retrieval: Recent Developments and Applications.* now, 2014.

[132] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1757–1772, July 2013.

[133] Bernhard Schölkopf, Alex J. Smola, Robert C. Williamson, and Peter L. Bartlett. New support vector algorithms. *Neural Computation*, 12(5):1207–1245, May 2000.

[134] C. Chandra Sekhar, Kazuya Takeda, and Fumitada Itakura. Recognition of consonant-vowel (cv) units of speech in a broadcast news corpus using support vector machines. In Seong-Whan Lee and Alessandro Verri, editors, *Pattern Recognition with Support Vector Machines: First International Workshop, SVM 2002 Niagara Falls, Canada, August 10, 2002 Proceedings*, pages 171–185. Springer Berlin Heidelberg, 2002.

[135] S. Sigtia, A. M. Stark, S. Krstulović, and M. D. Plumbley. Automatic environmental sound recognition: Performance versus computational cost. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(11):2096–2107, Nov 2016.

[136] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333, April 2018.

[137] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, January 2014.

[138] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M.D. Plumbley. Detection and classification of acoustic scenes and events. *Multimedia, IEEE Transactions on*, 17(10):1733–1746, Oct 2015.

[139] D. Stowell, M. Wood, Y. Stylianou, and H. Glotin. Bird detection in audio: A survey and a challenge. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, Sep. 2016.

[140] A. Stuhlsatz, C. Meyer, F. Eyben, T. Zielke, G. Meier, and B. Schuller. Deep neural networks for acoustic emotion recognition: Raising the benchmarks. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5688–5691, May 2011.

[141] A. Temko, E. Monte, and C. Nadeu. Comparison of sequence discriminant support vector machines for acoustic event classification. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 5, pages V–V, 2006.

[142] Andrey Temko, Robert Malkin, Christian Zieger, Dušan Macho, Climent Nadeu, and Maurizio Omologo. Clear evaluation of acoustic event detection and classification systems. In *Proceedings of the 1st International Evaluation Conference on Classification of Events, Activities and Relationships*, CLEAR'06, pages 311–322, Berlin, Heidelberg, 2007. Springer-Verlag.

[143] Andrey Temko, Climent Nadeu, and Joan-Isaac Biel. Acoustic event detection: Svm-based system and evaluation setup in clear'07. In Rainer Stiefelhagen, Rachel Bowers, and Jonathan Fiscus, editors, *Multimodal Technologies for Perception of Humans: International Evaluation Workshops CLEAR 2007 and RT 2007, Baltimore, MD, USA, May 8-11, 2007, Revised Selected Papers*, pages 354–363, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[144] Y. Tokozume and T. Harada. Learning environmental sounds with end-to-end convolutional neural network. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2721–2725, March 2017.

[145] H. D. Tran and H. Li. Sound event recognition with probabilistic distance svms. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(6):1556–1568, Aug 2011.

[146] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Semantic annotation and retrieval of music and sound effects. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):467–476, Feb 2008.

[147] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, July 2002.

[148] Zoltán Tüske, Pavel Golik, Ralf Schlüter, and Hermann Ney. Acoustic modeling with deep neural networks using raw time signal for lvcsr. In *[15th Annual Conference of the International Speech Communication Association, INTERSPEECH 2014, Singapore]*, pages 890–894, Sep 2014.

[149] J. Utans and J. Moody. Selecting neural network architectures via the prediction risk: application to corporate bond rating prediction. In *Proceedings First International Conference on Artificial Intelligence Applications on Wall Street*, pages 35–41, Oct 1991.

[150] M Vacher, D Istrate, Laurent Besacier, Jean-François Serignat, and Eric Castelli. Life Sounds Extraction and Classification in Noisy Environment. In *5th IASTED-SIP*, Hawaï, USA, United States, July 2003.

[151] Michel Vacher, Jean-François Serignat, Stéphane Chaillol, Dan Istrate, and Vladimir Popescu. Speech and sound use in a remote monitoring system for health care. In

Petr Sojka, Ivan Kopeček, and Karel Pala, editors, *Text, Speech and Dialogue*, pages 711–718, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[152] M. Valenti, S. Squartini, A. Diment, G. Parascandolo, and T. Virtanen. A convolutional neural network approach for acoustic scene classification. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 1547–1554, May 2017.

[153] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti. Scream and gunshot detection and localization for audio-surveillance systems. In *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 21–26, Sep. 2007.

[154] V. N. Vapnik and A. Ya. Chervonenkis. *Theory of Pattern Recognition [in Russian].* Nauka, USSR, 1974.

[155] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, Berlin, Heidelberg, 1995.

[156] R. Vergin, D. O'Shaughnessy, and A. Farhat. Generalized mel frequency cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition. *IEEE Transactions on Speech and Audio Processing*, 7(5):525–532, Sep. 1999.

[157] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 3637–3645, USA, 2016. Curran Associates Inc.

[158] Tuomas Virtanen, Rita Singh, and Bhiksha Raj. *Techniques for Noise Robustness in Automatic Speech Recognition.* Wiley Publishing, 1st edition, 2012.

[159] Lode Vuegen, Bert Van Den Broeck, Peter Karsmakers, Hugo Van hamme, and Bart Vanrumste. Automatic monitoring of activities of daily living based on real-life acoustic sensor data: a preliminary study. In *Proceedings of the Fourth Workshop on Speech and Language Processing for Assistive Technologies*, pages 113–118, Grenoble, France, August 2013. Association for Computational Linguistics.

[160] DeLiang Wang and Guy J. Brown. *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications.* Wiley-IEEE Press, 2006.

[161] Jia-Ching Wang, Jhing-Fa Wang, Kuok Wai He, and Cheng-Shu Hsu. Environmental sound classification using hybrid svm/knn classifier and mpeg-7 audio low-level descriptor. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pages 1731–1735, 2006.

[162] Z. Wang and I. Tashev. Learning utterance-level representations for speech emotion and age/gender recognition using deep neural networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5150–5154, March 2017.

[163] G. Wichern, J. Xue, H. Thornburg, B. Mechtley, and A. Spanias. Segmentation, indexing, and retrieval for environmental and natural sounds. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):688–707, March 2010.

[164] E. Wold, T. Blum, D. Keislar, and J. Wheaten. Content-based classification, search, and retrieval of audio. *IEEE MultiMedia*, 3(3):27–36, Fall 1996.

[165] Xu-Kui Yang, Liang He, Dan Qu, Wei-Qiang Zhang, and Michael T. Johnson. Semi-supervised feature selection for audio classification based on constraint compensated laplacian score. *EURASIP Journal on Audio, Speech, and Music Processing*, 2016(1):9, 2016.

[166] Jiaxing Ye, Takumi Kobayashi, Masahiro Murakawa, and Tetsuya Higuchi. Acoustic scene classification based on sound textures and events. In *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15, pages 1291–1294, New York, NY, USA, 2015. ACM.

[167] Antonio Torralba Yusuf Aytar, Carl Vondrick. Soundnet: Learning sound representations from unlabeled video. `https://github.com/cvondrick/soundnet`, 2016.

[168] H. Zhang, I. McLoughlin, and Y. Song. Robust sound event recognition using convolutional neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 559–563, April 2015.

[169] E. Çakir and T. Virtanen. End-to-end polyphonic sound event detection using convolutional recurrent neural networks with learned time-frequency representation input. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, July 2018.